

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-092753

(43)Date of publication of application : 28.03.2003

(51)Int.Cl.

H04N 7/24
G06F 12/00
G06F 17/30
H04N 7/08
H04N 7/081
// H04H 1/00

(21)Application number : 2001-222745

(71)Applicant : VICTOR CO OF JAPAN LTD

(22)Date of filing : 24.07.2001

(72)Inventor : ASAMI TOMOHIRO
ANDO ICHIRO

(30)Priority

Priority number : 2000388520 Priority date : 21.12.2000 Priority country : JP
2001044927 21.02.2001
2001207457 09.07.2001 JP

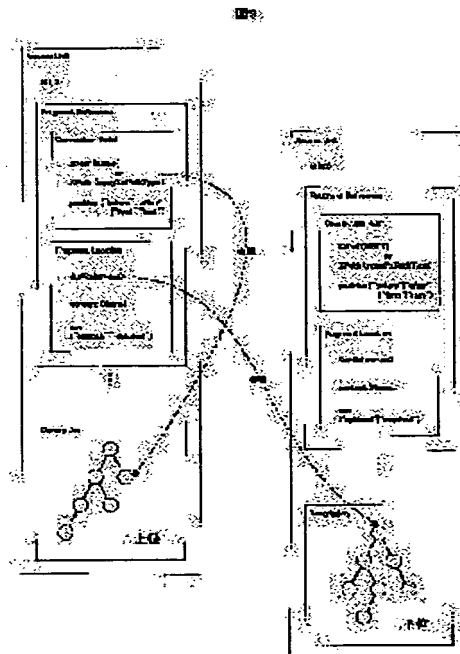
JP

(54) METHOD FOR TRANSMITTING STRUCTURED META-DATA

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method for transmitting structured meta-data for adopting a unit configuration of a plurality of structured data in an integrated way and transmitting each of the structured meta-data as each unit so as to enhance the transmission efficiency.

SOLUTION: Connection position information indicative of a separation point (that is, connecting point) of lower-rank structured meta-data in upper-rank structured meta-data and reference information for designating the separated lower-rank structured meta-data are attached to the separated upper-rank structured meta-data. Each of the separated structured meta-data is stored in a description being a meta-data storage section in an access unit. The connection position information and the reference information are stored in a Fragment Reference being a storage section of connection destination information. Each access unit is transmitted.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-92753

(P2003-92753A)

(43) 公開日 平成15年3月28日 (2003.3.28)

| (51) Int.Cl. ⁷ | 識別記号 | F I | テ-マコード (参考) |
|---------------------------|-------|---------------|-------------------|
| H 0 4 N 7/24 | | G 0 6 F 12/00 | 5 4 7 H 5 B 0 7 5 |
| G 0 6 F 12/00 | 5 4 7 | 17/30 | 1 1 0 F 5 B 0 8 2 |
| 17/30 | 1 1 0 | | 1 7 0 G 5 C 0 5 9 |
| | 1 7 0 | | 2 2 0 C 5 C 0 6 3 |
| | 2 2 0 | H 0 4 H 1/00 | C |

審査請求 未請求 請求項の数27 O L (全 35 頁) 最終頁に続く

| | | | |
|--------------|------------------------------|----------|--|
| (21) 出願番号 | 特願2001-222745 (P2001-222745) | (71) 出願人 | 000004329 日本ビクター株式会社 神奈川県横浜市神奈川区守屋町3丁目12番地 |
| (22) 出願日 | 平成13年7月24日 (2001.7.24) | (72) 発明者 | 浅見 知弘 神奈川県横浜市神奈川区守屋町3丁目12番地 日本ビクター株式会社内 |
| (31) 優先権主張番号 | 特願2000-388520 (P2000-388520) | (72) 発明者 | 安藤 一郎 神奈川県横浜市神奈川区守屋町3丁目12番地 日本ビクター株式会社内 |
| (32) 優先日 | 平成12年12月21日 (2000.12.21) | | |
| (33) 優先権主張国 | 日本 (J P) | | |
| (31) 優先権主張番号 | 特願2001-44927 (P2001-44927) | | |
| (32) 優先日 | 平成13年2月21日 (2001.2.21) | | |
| (33) 優先権主張国 | 日本 (J P) | | |
| (31) 優先権主張番号 | 特願2001-207457 (P2001-207457) | | |
| (32) 優先日 | 平成13年7月9日 (2001.7.9) | | |
| (33) 優先権主張国 | 日本 (J P) | | |

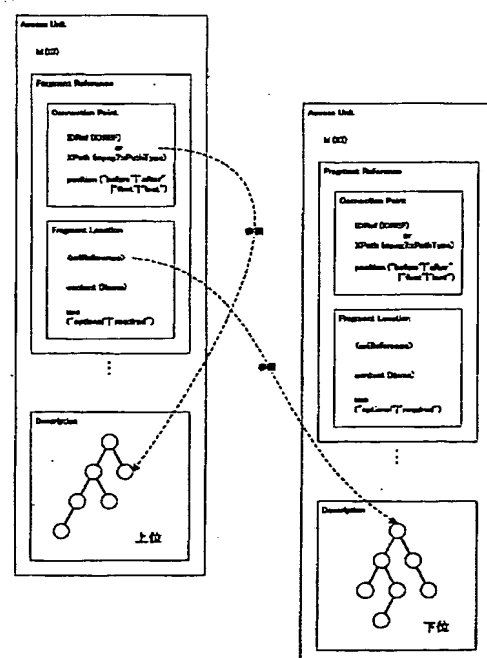
最終頁に続く

(54) 【発明の名称】 構造化メタデータの伝送方法

(57) 【要約】

【課題】 統合可能なように複数の構造化メタデータをユニット化し、各構造化メタデータを各ユニットとして伝送することにより伝送効率を高めることを可能とする構造化メタデータの伝送方法を提供する。

【解決手段】 分離した上位の構造化メタデータに、上位の構造メタデータ中の下位の構造化メタデータの分離点 (即ち接続点) を示す接続位置情報と、分離した下位の構造化メタデータを指定する参照情報とを付加する。分割された各構造化メタデータは Access Unit 内のメタデータ格納部である Description に格納される。接続位置情報と参照情報とは接続先情報格納部である Fragment Reference に格納される。この各 Access Unit を伝送する。



【特許請求の範囲】

【請求項1】コンテンツの内容を記述するための第1の構造化メタデータとその第1の構造化メタデータに対して下位の関係となる第2の構造化メタデータとを伝送するための構造化メタデータの伝送方法であって、メタデータ格納部と接続先情報格納部とを有し、前記メタデータ格納部には前記第1の構造化メタデータが格納され、前記接続先情報格納部には前記第1の構造化メタデータ内における前記第2の構造化メタデータの接続点を示す接続位置情報と、少なくとも前記第2の構造化メタデータの存在場所に関係する情報を有する前記第2の構造化メタデータに関する参照情報とが格納される第1のユニットと、

前記第2の構造化メタデータを格納したメタデータ格納部を有する第2のユニットとを用い、

前記各ユニットはそれぞれのユニットを特定するための識別情報を備えて伝送されることを特徴とする構造化メタデータの伝送方法。

【請求項2】請求項1記載の構造化メタデータの伝送方法において、

前記各ユニットの伝送頻度をユニット単位で制御することを特徴とする構造化メタデータの伝送方法。

【請求項3】請求項1または2記載の構造化メタデータの伝送方法において、

前記接続位置情報は、前記第1の構造化メタデータ内におけるノードを指定する情報と、そのノードに対する前記第2の構造化メタデータの接続位置を指定する情報とを有することを特徴とする構造化メタデータの伝送方法。

【請求項4】請求項1～3のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第2の構造化メタデータに関する参照情報は、前記第2の構造化メタデータの内容に関する情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項5】請求項1～4のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第2の構造化メタデータに関する参照情報は、前記第1の構造化メタデータへの統合の必要性を示す情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項6】請求項1～5のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第2の構造化メタデータに関する参照情報が有する前記第2の構造化メタデータの存在場所に関係する情報は、前記第2のユニットに付加された前記ユニットを特定するための識別情報に基づく情報であることを特徴とする構造化メタデータの伝送方法。

【請求項7】請求項1～6のいずれか一つに記載の構造化メタデータの伝送方法において、

前記ユニットを特定するための識別情報、前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接

続位置情報、及び前記第2の構造化メタデータに関する参照情報は、テキスト形式で記述されていることを特徴とする構造化メタデータの伝送方法。

【請求項8】請求項1～6のいずれか一つに記載の構造化メタデータの伝送方法において、

前記ユニットを特定するための識別情報、前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接続位置情報、及び前記第2の構造化メタデータに関する参照情報は、バイナリ形式で記述されていることを特徴とする構造化メタデータの伝送方法。

【請求項9】請求項1～8のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第1及び第2の構造化メタデータを、電子番組案内を記述するための木構造を有する構造化メタデータとしたことを特徴とする構造化メタデータの伝送方法。

【請求項10】請求項1～9のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第1の構造化メタデータと前記第2の構造化メタデータとは、木構造を有する1つの元の構造化メタデータから分割されたものであり、

前記接続位置情報は、前記元の構造化メタデータにおける、前記第2の構造化メタデータの最上位ノードに相当するノードの名前に関する名前情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項11】請求項1～9のいずれか一つに記載の構造化メタデータの伝送方法において、

前記接続位置情報は、前記第1の構造化メタデータと前記第2の構造化メタデータとが統合された場合に生成されると想定した木構造を有する第3の構造化メタデータにおける、前記第2の構造化メタデータの最上位ノードに相当するノードの名前に関する名前情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項12】請求項1～11のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第2の構造化メタデータが2つ以上存在し、全ての前記第2の構造化メタデータを格納した1つの第2のユニット、及び／または、1つの前記第2の構造化メタデータを最小単位として、前記分割した全ての第2の構造化メタデータの内の一部分を格納した複数の第2のユニットを用い、

各前記第2の構造化メタデータの各接続位置情報における2つ以上の前記接続位置情報に共通する情報である共通接続位置情報と、各前記第2の構造化メタデータに関する各参照情報における2つ以上の前記参照情報に共通する情報である共通参照情報との、2つの共通情報の内の少なくとも一方の共通情報を、前記第1のユニット内の接続先情報格納部に属性共通情報として有することを特徴とする構造化メタデータの伝送方法。

【請求項13】コンテンツの内容を記述するための第1の構造化メタデータと、その第1の構造化メタデータに

対して下位の関係となる第2の構造化メタデータとを伝送するための構造化メタデータの伝送方法であって、前記第1の構造化メタデータと前記第2の構造化メタデータとは、木構造を有する1つの元の構造化メタデータから分割されたものであり、

メタデータ格納部と接続先情報格納部とを有し、前記メタデータ格納部には前記第1の構造化メタデータが格納され、前記接続先情報格納部には、前記元の構造化メタデータにおける前記第2の構造化メタデータの最上位ノードに相当するノードを示す接続位置情報と、少なくとも前記第2の構造化メタデータの存在場所に関する情報を有する前記第2の構造化メタデータに関する参照情報とが格納された第1のユニットと、前記第2の構造化メタデータが格納されたメタデータ格納部を有する第2のユニットと、を伝送することを特徴とする構造化メタデータの伝送方法。

【請求項14】コンテンツの内容を記述するための第1の構造化メタデータとその第1の構造化メタデータに対して下位の関係となる第2の構造化メタデータとを伝送するための構造化メタデータの伝送方法であって、メタデータ格納部と接続先情報格納部とを有し、前記メタデータ格納部には前記第1の構造化メタデータが格納され、前記接続先情報格納部には、前記第1の構造化メタデータと前記第2の構造化メタデータとが統合された場合に生成されると想定した木構造を有する第3の構造化メタデータにおける、前記第2の構造化メタデータの最上位ノードに相当するノードを示す接続位置情報と、少なくとも前記第2の構造化メタデータの存在場所に関する情報を有する前記第2の構造化メタデータに関する参照情報とが格納された第1のユニットと、前記第2の構造化メタデータが格納されたメタデータ格納部を有する第2のユニットと、を伝送することを特徴とする構造化メタデータの伝送方法。

【請求項15】請求項13に記載の構造化メタデータの伝送方法において、前記接続位置情報は、前記元の構造化メタデータにおける、前記第2の構造化メタデータの最上位ノードに相当するノードの名前に関する名前情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項16】請求項14に記載の構造化メタデータの伝送方法において、前記接続位置情報は、前記第3の構造化メタデータにおける、前記第2の構造化メタデータの最上位ノードに相当するノードの名前に関する名前情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項17】請求項13～16のいずれか一つに記載の構造化メタデータの伝送方法において、前記各ユニットの伝送頻度をユニット単位で制御することを特徴とする構造化メタデータの伝送方法。

【請求項18】請求項13～17のいずれか一つに記載

の構造化メタデータの伝送方法において、

前記第2の構造化メタデータに関する参照情報は、前記第2の構造化メタデータの内容に関する情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項19】請求項13～18のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第2の構造化メタデータに関する参照情報は、前記第1の構造化メタデータへの統合の必要性を示す情報を有することを特徴とする構造化メタデータの伝送方法。

【請求項20】請求項13～19のいずれか一つに記載の構造化メタデータの伝送方法において、

前記各ユニットはそれぞれのユニットを特定するための識別情報を備えて伝送されることを特徴とする構造化メタデータの伝送方法。

【請求項21】請求項20に記載の構造化メタデータの伝送方法において、

前記第2の構造化メタデータに関する参照情報が有する前記第2の構造化メタデータの存在場所に関する情報は、前記第2のユニットに付加された前記ユニットを特定するための識別情報に基づく情報であることを特徴とする構造化メタデータの伝送方法。

【請求項22】請求項13～19のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接続位置情報、及び前記第2の構造化メタデータに関する参照情報は、テキスト形式で記述されていることを特徴とする構造化メタデータの伝送方法。

【請求項23】請求項20または21に記載の構造化メタデータの伝送方法において、

前記ユニットを特定するための識別情報、前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接続位置情報、及び前記第2の構造化メタデータに関する参照情報は、テキスト形式で記述されていることを特徴とする構造化メタデータの伝送方法。

【請求項24】請求項13～19のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接続位置情報、及び前記第2の構造化メタデータに関する参照情報は、バイナリ形式で記述されていることを特徴とする構造化メタデータの伝送方法。

【請求項25】請求項20または21に記載の構造化メタデータの伝送方法において、

前記ユニットを特定するための識別情報、前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接続位置情報、及び前記第2の構造化メタデータに関する参照情報は、バイナリ形式で記述されていることを特徴とする構造化メタデータの伝送方法。

【請求項26】請求項13～25のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第1及び第2の構造化メタデータを、電子番組案内

を記述するための構造化メタデータとしたことを特徴とする構造化メタデータの伝送方法。

【請求項27】請求項13～26のいずれか一つに記載の構造化メタデータの伝送方法において、

前記第2の構造化メタデータが2つ以上存在し、
全ての前記第2の構造化メタデータを格納した1つの第2のユニット、及び／または、1つの前記第2の構造化メタデータを最小単位として、前記分割した全ての第2の構造化メタデータの内の一部分を格納した複数の第2のユニットを用い、

各前記第2の構造化メタデータの各接続位置情報における2つ以上の前記接続位置情報に共通する情報である共通接続位置情報と、各前記第2の構造化メタデータに関する各参照情報における2つ以上の前記参照情報に共通する情報である共通参照情報との、2つの共通情報の内の少なくとも一方の共通情報を、前記第1のユニット内の接続先情報格納部に属性共通情報として有することを特徴とする構造化メタデータの伝送方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、デジタルTV放送等におけるコンテンツ内容を記述した構造化メタデータの伝送方法に関するものである。

【0002】

【従来の技術】AVコンテンツの内容を記述するための枠組みとして、MPEG-7の標準化作業が進められている。MPEG-7とは、AVコンテンツを時間的及び空間的に構造化し、AVコンテンツの内容を木構造を用いて記述する構造化メタデータの規格である。

【0003】MPEG-7記述は、Descriptor (D)とDescription Scheme (DS)により構成される。Dはコンテンツの特徴を記述するための記述子であり、DSは構造を記述するためのもの（内容記述を記述子間の関係によって構成する方法の規定）である。DSは、Dやより下位のDSを要素として持つことができ、記述子群に相当する。また、DやDSのシンタックス（構文）を規定するための言語が、Description Definition Language (DDL)である。

【0004】MPEG-7記述のフォーマットには、XML (Extensible Markup Language)を用いる。また、DDLはXML SchemaにMPEG-7独自の拡張を加えたものである。

【0005】

【発明が解決しようとする課題】AVコンテンツ内容の検索の高速化、効率化のためには、前述した木構造を用いてコンテンツ内容を記述した構造化メタデータが必要である。

【0006】しかし、現段階におけるMPEG-7標準化作業では、1つのMPEG-7記述は1つの文書で構

成されており、そのため、例えばデジタルTV放送において、1チャンネル内の1日分の放送内容のMPEG-7記述（構造化メタデータ）を作るとその文書は非常に巨大になり、伝送効率が悪い。

【0007】伝送効率を上げるために単純にMPEG-7記述の文書を分割したのでは、受信側で分割された文書を統合して元の構造化メタデータを再現できない。単なる分割、統合ではMPEG-7記述が断片情報となってしまう、コンテンツの内容記述情報として使用できなくなる。さらに、単なる分割、統合では、上位階層の情報、下位階層の情報といった、コンテンツの内容記述の詳細度に応じた分割、統合ができない。

【0008】また、受信不良や視聴者によるチャンネルの切り替え等によりMPEG-7記述に欠落情報が発生した場合、欠落情報部分を部分的に再度補完することができず不便であった。

【0009】本発明は、統合可能なように複数の構造化メタデータをユニット化し、各構造化メタデータを各ユニットとして伝送することにより伝送効率を高めることを可能とする構造化メタデータの伝送方法を提供することを目的としている。さらに本発明は、受信側において欠落データが発生した場合にも、不完全となる構造化メタデータの大きさを小さくとどめることができると共に、不完全な構造化メタデータのみを再度補間可能とする構造化メタデータの伝送方法を提供することを目的としている。

【0010】

【課題を解決するための手段】そこで、上記目的を達成するために本発明は、下記の方法を提供するものである。

(1) コンテンツの内容を記述するための第1の構造化メタデータとその第1の構造化メタデータに対して下位の関係となる第2の構造化メタデータとを伝送するための構造化メタデータの伝送方法であって、メタデータ格納部と接続先情報格納部とを有し、前記メタデータ格納部には前記第1の構造化メタデータが格納され、前記接続先情報格納部には前記第1の構造化メタデータ内における前記第2の構造化メタデータの接続点を示す接続位置情報と、少なくとも前記第2の構造化メタデータの存在場所に関する情報を有する前記第2の構造化メタデータに関する参照情報とが格納される第1のユニットと、前記第2の構造化メタデータを格納したメタデータ格納部を有する第2のユニットとを用い、前記各ユニットにそれぞれのユニットを特定するための識別情報を付加して前記各ユニットを伝送することを特徴とする構造化メタデータの伝送方法。

(2) コンテンツの内容を記述するための第1の構造化メタデータと、その第1の構造化メタデータに対して下位の関係となる第2の構造化メタデータとを伝送するための構造化メタデータの伝送方法であって、前記第1

の構造化メタデータと前記第2の構造化メタデータとは、木構造を有する1つの元の構造化メタデータから分割されたものであり、メタデータ格納部と接続先情報格納部とを有し、前記メタデータ格納部には前記第1の構造化メタデータが格納され、前記接続先情報格納部には、前記元の構造化メタデータにおける前記第2の構造化メタデータの最上位ノードに相当するノードを示す接続位置情報と、少なくとも前記第2の構造化メタデータの存在場所に関する情報を有する前記第2の構造化メタデータに関する参照情報とが格納された第1のユニットと、前記第2の構造化メタデータが格納されたメタデータ格納部を有する第2のユニットと、を伝送することを特徴とする構造化メタデータの伝送方法。

(3) コンテンツの内容を記述するための第1の構造化メタデータとその第1の構造化メタデータに対して下位の関係となる第2の構造化メタデータとを伝送するための構造化メタデータの伝送方法であって、メタデータ格納部と接続先情報格納部とを有し、前記メタデータ格納部には前記第1の構造化メタデータが格納され、前記接続先情報格納部には、前記第1の構造化メタデータと前記第2の構造化メタデータとが統合された場合に生成されると想定した木構造を有する第3の構造化メタデータにおける、前記第2の構造化メタデータの最上位ノードに相当するノードを示す接続位置情報と、少なくとも前記第2の構造化メタデータの存在場所に関する情報を有する前記第2の構造化メタデータに関する参照情報とが格納された第1のユニットと、前記第2の構造化メタデータが格納されたメタデータ格納部を有する第2のユニットと、を伝送することを特徴とする構造化メタデータの伝送方法。

【0011】

【発明の実施の形態】本発明の実施例においては、元の構造化メタデータ（ここではMPEG-7記述）を任意の位置で分離する（あるいは上位の構造化メタデータとそれに統合されるべき下位の構造化メタデータとを作成する）と共に、分離した上位の構造化メタデータに、下位の構造化メタデータの分離点（即ち接続点）を示す接続位置情報と、分離した下位の構造化メタデータを指定する参照情報とを付加するものである。この参照情報は、少なくとも第2の構造化メタデータの存在場所に関する情報を有する第2の構造化メタデータに関する参照情報である。参照情報に、下位の構造化メタデータの内容や処理方法等に関する情報をも記述し、ユーザーやアプリケーションが下位の構造化メタデータを処理するための判断材料とするようにしてもよい。

【0012】また、本実施例は、上位の構造化メタデータに付加された分離点（接続点）を示す接続位置情報と、下位の構造化メタデータを指定する参照情報とに基づき、下位の構造化メタデータを上位の構造化メタデータに接続し、元の構造化メタデータを再構成する（ある

いは上位の構造化メタデータとそれに統合されるべき下位の構造化メタデータとを統合する）ものである。

【0013】図1は、本発明の第1実施例を用いたMPEG-7記述（構造化メタデータ）の分割、蓄積・伝送、再構成（統合）方法の内の、分割から蓄積・伝送までの処理の概略を示す図である。MPEG-7記述は複数のMPEG-7記述に分割され、分割された各MPEG-7記述はAccess Unit内のメタデータ格納部に格納される。インターネット等の蓄積系システムにおいては、1つまたは複数のAccess Unitがファイル形式で蓄積される。一方、ディジタル・テレビジョン放送等の伝送系システムにおいては、複数のAccess Unitを連続的に並べ、データ・ストリームを構成して伝送される。

【0014】図2は、本発明の第1実施例を用いたMPEG-7記述の分割、蓄積・伝送、再構成（統合）方法の内の、蓄積・伝送から再構成までの処理の概略を示す図である。蓄積系システムまたは伝送系システムから、複数のAccess Unitを取得する。複数のAccess Unit内のメタデータ格納部に格納されているそれぞれのMPEG-7記述を再構成することにより、元のMPEG-7記述を得る。

【0015】図3は、本発明の第1実施例に用いるAccess Unitの構成例を示す概念図である。また、図4は、図3に示したAccess UnitのDDLシンタックス例である。

【0016】図3に示すAccess Unitは、メタデータ格納部であるDescriptionと、接続先情報格納部であるFragment Referenceと、属性id（識別情報）格納部とを備えている。

【0017】Descriptionは、元のMPEG-7記述から分割された各MPEG-7記述を格納するものである。Descriptionは、1つのAccess Unit中に1つだけ存在する。

【0018】Fragment Referenceは、Descriptionに格納されるMPEG-7記述とその下位のMPEG-7記述とを再接続する際に参照する接続先情報（接続位置情報と参照情報）を格納するものである。Descriptionに格納される上位MPEG-7記述に接続されるべき下位のMPEG-7記述が複数存在する場合、Fragment Referenceも下位のMPEG-7記述毎にそれぞれ存在する。また、Descriptionに格納される上位MPEG-7記述に接続されるべき下位のMPEG-7記述が1つも存在しない場合、Fragment Referenceも1つも存在しない。

【0019】Fragment Referenceは、Connection PointとFragment Locationの2つの子要素を備えている。

【0020】Connection Pointは、Descriptionに格納される上位MPEG-7記述中の、下位MPEG-7記述を接続する場所（接続点）を示す接続位置情報である。IDRefまたはXPathのいずれかの子供要素を用いて、上位MPEG-7記述中の特定のノードを指定する。IDRefは、MPEG-7記述中に記述されたIDによって特定のノードを指定するものである。一方、XPathは、W3C (World Wide Web Consortium) で定められたXMLパス言語XPath (<http://www.w3.org/TR/xpath>) を用いて、MPEG-7記述中の特定のノードを指定するものである。

【0021】さらに、Connection Point内のposition (属性position) は、IDRefまたはXPathによって指定されたノードに対して、下位MPEG-7記述を接続する位置を指定する。positionの値がbeforeの場合は指定されたノードの前に接続し、afterの場合は指定されたノードの後に接続する。firstの場合は指定されたノードの子ノードの先頭に接続し、lastの場合は指定されたノードの子ノードの最後に接続する。

【0022】この属性positionの使用方法について、図8から図11を用いて説明する。図8から図11の各図では、上位MPEG-7記述と下位MPEG-7記述、及び接続後のMPEG-7記述が、模式図と実際のXML記述とによって表されている。

【0023】図8では、Connection Pointの要素IDRefまたはXPathによって上位MPEG-7記述のノードBが指定され、属性positionの値がbeforeである。このとき、下位MPEG-7記述はノードBの前に接続される。

【0024】図9では、Connection Pointの要素IDRefまたはXPathによって上位MPEG-7記述のノードBが指定され、属性positionの値がafterである。このとき、下位MPEG-7記述はノードBの後に接続される。

【0025】図10では、Connection Pointの要素IDRefまたはXPathによって上位MPEG-7記述のノードAが指定され、属性positionの値がfirstである。このとき、下位MPEG-7記述はノードAの子供要素の先頭、即ちノードBの前に接続される。

【0026】図11では、Connection Pointの要素IDRefまたはXPathによって上位MPEG-7記述のノードAが指定され、属性positionの値がlastである。このとき、下位MPEG-7記述はノードAの子供要素の最後、即ちノードBの後に接続される。

【0027】図3にもどって、Fragment Re

ferenceのもう一方の子供要素であるFragment Locationは、Descriptionに格納される上位MPEG-7記述に接続する下位MPEG-7記述を示す参照情報である。Fragment Locationのデータ型はuriReferenceであり、URI (Uniform Resource Identifiers) によって下位MPEG-7記述のルート要素を参照する（下位MPEG-7記述の存在場所を指定する。）。ここでは、URIに加え、W3Cで定められたXMLポイント言語XPointer (<http://www.w3.org/TR/xptr>) を利用し、外部XML文書内部の特定の要素を参照する。

【0028】また、Fragment Locationはcontent (属性content) とuse (属性use) とを持つ。属性contentは下位MPEG-7記述の内容の概略を記述するものである。ここでは、例えば、下位MPEG-7記述のルート要素名が使用できる。この属性を利用し、ユーザーまたはアプリケーションは、必要なMPEG-7記述を選択的に入手または保存することができる。属性useは下位MPEG-7記述の必要性を示すものである。デフォルト値はoptionalであるが、requiredと指定することにより、MPEG-7アプリケーションに下位MPEG-7記述を入手または保存することを強制することができる。下位MPEG-7記述を著作権保護データとしておけば、上位MPEG-7記述の著作権保護を強化できる。

【0029】属性id格納部は、複数のAccess Unitを一意に識別するためのid (属性id: 識別情報) を格納している。インターネット等の蓄積系システムの場合は、Access Unitはファイルに格納され、URIを用いてAccess Unitを一意に識別することができる。しかし、デジタル・テレビジョン放送等の伝送系システムの場合、複数のAccess Unitを連続的に並べてデータ・ストリームが構成されるので、インターネットと同様のURIを用いることができない。そこで、Access Unitの属性idを利用してAccess Unitを一意に識別する。

【0030】以上の構成により、複数のAccess Unitを取得したとき、接続先情報格納部であるFragment Referenceに格納された情報（接続位置情報と参照情報）を元にして、分離された上位MPEG-7記述と下位MPEG-7記述とを順次接続することにより、元のMPEG-7記述を再構成することができる。また、必要なAccess Unit、即ち必要なMPEG-7記述を参照情報に基づき選択することにより、元のMPEG-7記述を部分的に再構成することもできる。さらには、その伝送において、分割されたMPEG-7記述を各Access Unitと

して伝送することにより、各Access Unit毎の伝送制御が行えるようになり伝送効率を高めることが可能となる。

【0031】また、接続のための情報である接続位置情報と参照情報とを、メタデータ格納部(Description)とは異なる接続先情報格納部(FragmentReference)に格納するようにしたので、構造化メタデータの内部を操作することなく接続のための情報を付加することができるという効果がある。

【0032】次に、本発明の第2実施例について説明する。第2実施例では、MPEG-7記述の分割から蓄積・伝送までの処理の流れと、蓄積・伝送から再構成までの処理の流れは第1実施例と同様である。即ち、図1及び図2に示す処理の流れは、第2実施例にも当てはまる。但し、第2実施例では、Access Unitの構成が第1実施例のそれとは異なる。

【0033】図5は、本発明の第2実施例に用いる、Access Unitの他の構成例を示す概念図である。また、図6、図7は、図5に示したAccess UnitのDDLシンタックスである。

【0034】本発明の第2実施例では、Access Unitは接続先情報格納部であるFragmentReference、メタデータ格納部を有するUpdate Command、及び属性id(識別情報)格納部を備えている。

【0035】FragmentReferenceの構成は、第1実施例におけるFragmentReferenceの構成と同様である。FragmentReferenceは1つのAccess Unit中に複数存在する場合もあるし、1つも存在しない場合もある。

【0036】また、属性id(識別情報)格納部についても、本発明の第1の実施例と同様である。

【0037】Update Commandは、コマンドの実行によりMPEG-7記述を構成するものであり、Type、Location、Valueの3つの子供要素により構成される。Update Commandは1つのAccess Unit中に1つ以上存在する。

【0038】Typeは実行するコマンドの種類であり、add、delete、change、resetのいずれかの値をとる。Typeの値がaddの場合は、Locationによって示されるMPEG-7記述中の特定の位置に、Valueに格納されるMPEG-7記述を接続する。Typeの値がdeleteの場合は、Locationによって示されるMPEG-7記述中のノードとそれ以下のノードを削除する。Typeの値がchangeの場合は、Locationによって示されるMPEG-7記述中のノードとそれ以下のノードを、Valueに格納されるMPEG-7記述に

置き換える。Typeの値がresetの場合は、MPEG-7記述を予め定められた初期状態に戻す。

【0039】Locationは、コマンドを実行する場所を示す位置情報である。Locationのデータ型はuriReferenceであり、URI及びXPath pointerによってMPEG-7記述中のコマンド実行対象のノードを参照する。ここで参照するMPEG-7記述は、これ以前のコマンド実行によりすでに生成されているMPEG-7記述である。また、Locationは属性positionを持つ。属性positionはTypeの値がaddである場合のみ有効であり、Locationによって参照されるノードに対して、Valueに格納されるMPEG-7記述を接続する位置を指定する。positionの値がbeforeの場合は指定されたノードの前に接続し、afterの場合は指定されたノードの後に接続する。firstの場合は指定されたノードの子ノードの先頭に接続し、lastの場合は指定されたノードの子ノードの最後に接続する。

【0040】Valueは、MPEG-7記述を格納するためのものである。

【0041】Update Commandにおいて、Typeの値がaddである場合の動作は、複数のMPEG-7記述を接続して1つのMPEG-7記述を再構成するものであるので、FragmentReferenceの持つ機能と類似している。両者の機能の違いは、FragmentReferenceによるMPEG-7記述の再構成が、上位MPEG-7記述から下位MPEG-7記述への参照情報を利用し、Access Unit群を取得(または受信)したユーザー(またはアプリケーション)の意図により行われるのに対し、Update CommandによるMPEG-7記述の再構成は、下位MPEG-7記述から上位MPEG-7記述への参照情報を利用し、Access Unit群の作成者または送信者の意図により行われることである。即ち、FragmentReferenceの場合は、ユーザーまたはアプリケーションがFragmentLocationに記述された情報を参考にして必要なMPEG-7記述を選択して元のMPEG-7記述を部分的に再構成できるのに対し、Update Commandの場合は、MPEG-7記述の作成者または送信者がユーザーの側に再構成されるMPEG-7記述を制御することができる。従って、両者の機能は異なり、目的に応じて使い分けるものである。

【0042】次に、図8～図11で説明したConnection Point(接続位置情報)内の属性position(指定されたノードに対する下位MPEG-7記述の接続位置を指定する情報)について他の例を説明する。図8～図11では属性positionの値が4つの値の場合を示したが、この内の2値を用いるこ

と(beforeとlast、またはafterとfirst)によって、任意の接続位置を指定することも可能である。ここでは、属性positionの取り得る値が、beforeAsSiblingとlastAsChildの場合について説明する。

【0043】beforeAsSiblingは、Fragment Location (下位MPEG-7記述に関する参照情報) により指定された下位MPEG-7記述を、Connection Pointにより指定されたノードの前に兄弟として付加することを意味する。また、lastAsChildは、Fragment Locationにより指定された下位MPEG-7記述を、Connection Pointにより指定されたノードの子供の最後に付加することを意味する。

【0044】この属性positionの使用方法について、図12と図13を用いて説明する。図12と図13では、上位MPEG-7記述と下位MPEG-7記述、及び接続後のMPEG-7記述が、模式図と実際のXML記述とによって表されている。

【0045】図12では、Connection Pointの要素IDRefまたはXPathによって上位MPEG-7記述のノードBが指定され、属性positionの値がbeforeAsSiblingである。このとき、下位MPEG-7記述はノードBの前に接続される。

【0046】図13では、Connection Pointの要素IDRefまたはXPathによって上位MPEG-7記述のノードAが指定され、属性positionの値がlastAsChildである。このとき、下位MPEG-7記述はノードAの子供要素の最後、即ちノードBの後に接続される。

【0047】次に、この2値によって任意の位置を指定できることを説明する。あるノードAが子供を持たず、そのノードの子供として下位MPEG-7記述追加する場合、Connection PointでノードAを指定し、属性positionをlastAsChildとすればよい。あるノードAが子供を持ち、そのノードの子供の最後に下位MPEG-7記述追加する場合、Connection PointでノードAを指定し、属性positionをlastAsChildとすればよい。あるノードAが子供を持ち、そのノードの子供の最後以外の位置(即ちノードAの子供ノードBの前の位置)に下位MPEG-7記述追加する場合、Connection PointでノードBを指定し、属性positionをbeforeAsSiblingとすればよい。《バイナリ・フォーマット》ここまでの説明では、Access Unit内のMPEG-7記述のデータ及び各情報をテキスト・データで表現したものとしたが、次に、Access Unit内のMPEG-7

記述のデータ及び各情報をバイナリ・データで表現した第3実施例について説明する。

【0048】テキスト・データで表現された各情報は、ユーザーにとって理解が容易である、編集が容易である、インターネットでのデータの送受信に好適である等の利点を有する。一方、バイナリ・データで表現された各情報は、テキスト・データで表現された各情報よりも情報量が少なく伝送効率を向上させることができるので、ディジタル・テレビジョン放送、携帯端末向けのデータ配信等でのデータの伝送に好適である。

【0049】図14は、本発明の第3実施例のAccess Unitのバイナリ・フォーマットを示す図である。

【0050】AUIDは各Access Unitを一意に識別するためのIDである。全てのAccess Unitに一意の番号を割り振る。

【0051】Length of fragment referenceは、Length of fragment referenceの終わりからSub-tree binary representationの初めまでの長さをビット単位で表す。これにより、Fragment referenceをデコードすることなく、Sub-tree binary representationに高速にアクセスすることができる。

【0052】Number of fragment referenceは、その後続くFragment reference (接続位置情報と参照情報) の数を表す。

【0053】Fragment referenceは、上位の構造化メタデータと下位の構造化メタデータを接続するための情報(即ち接続先情報である接続位置情報と参照情報)である。1つのAccess Unitに複数のFragment referenceを記述することができる。Fragment referenceの詳細なフォーマットについては、後で説明する。

【0054】Sub-tree binary representationは、構造化メタデータをバイナリ・データで表現したものである。例えば、XMLで記述された構造化メタデータを、何らかのバイナリ符号化方式によりバイナリ・データに変換する。XMLのバイナリ符号化方式としては、WAP (Wireless Application Protocol) Forumで規定するBinary XML Content Format Specification等がある。

【0055】図15は、Fragment referenceのバイナリ・フォーマットを示す図である。

【0056】Reference of Contents IDは、参照する下位構造化メタデータの内容を、内容の種類ごとに一意に割り振られたIDによって

示すものである（下位構造化メタデータの内容に関する情報である）。予め構造化メタデータの内容を分類し、各分類に対してIDを一意に割り振っておく必要がある。

【0057】Necessityは、参照する下位構造化メタデータの必要性を1ビットで表すものである（上位構造化メタデータへの統合の必要性を示す情報である）。0ならばオプション、1ならば必須である。

【0058】Reference to AU IDは、参照する下位構造化メタデータを格納するAccess Unitに付加されたAU IDを用いて下位構造化メタデータを参照するための情報（下位構造化メタデータの存在場所に関する情報）である。

【0059】上記のReference of Contents ID、Necessity、及びReference to AU IDの3つが、下位構造化メタデータに関する参照情報である。

【0060】Connection pointは、Sub-tree binary representationに格納される構造化メタデータ中の下位構造化メタデータの接続点を示す接続位置情報である。ここでは、例えばXPathによって構造化メタデータ内の任意のノードを指定し、positionによって、その指定されたノードに対しての下位構造化メタデータを接続する位置を指定する。さらに、XPathとpositionを何らかのバイナリ符号化方式でバイナリデータに変換したものを用いてもよい。《分割ファイル伝送のためのシステム》図16は、ディジタル・テレビジョン放送等でビデオ・コンテンツと共にMPEG-7記述を伝送するためのMPEG-7伝送システムを示す図である。

【0061】XML文書として記述されたMPEG-7記述は、分割器によって、予め設定された基準に従って適当な数の断片に分割される。さらに、各断片に再統合のために必要なFragment Reference、属性id等が付加され、図3、5に示すようなAccess Unitが構成される。Access UnitはMPEG-7バイナリ符号化器へ出力される。

【0062】MPEG-7バイナリ符号化器は、入力されるMPEG-7記述を有するAccess Unitをバイナリに符号化する。符号化されたバイナリMPEG-7 Access Unitは、伝送制御器へ出力される。

【0063】一方、ビデオ・コンテンツは、ビデオ符号化器によって符号化される。ここではMPEG-2等のビデオ符号化方式を用いる。ビデオ符号化データは、伝送制御器へ出力される。

【0064】伝送制御器は、入力されるバイナリMPEG-7 Access Unitとビデオ符号化データをマルチプレクスし、伝送する。バイナリMPEG-7

Access Unitは、カルーセル方式で伝送される。その際、あらかじめ設定された基準に従って、カルーセル伝送の期間や頻度をAccess Unit毎に制御する。例えば、重要な情報を含むAccess Unitほど、また、上位構造化メタデータを有するAccess Unitほど、カルーセル伝送の期間を長くし、伝送頻度を高くする。

【0065】図17は、図16に示すMPEG-7伝送システムにより伝送される放送データを受信するためのMPEG-7受信システムを示す図である。

【0066】受信された放送データは、受信機によりデマルチプレクスされ、バイナリMPEG-7 Access Unitとビデオ符号化データとに分けられる。バイナリMPEG-7 Access UnitはMPEG-7バイナリ復号化器へ出力され、ビデオ符号化データはビデオ復号化器へ出力される。

【0067】MPEG-7バイナリ復号化器は、入力されるバイナリMPEG-7 Access Unitを複合化し、XML形式のMPEG-7 Access Unit（テキスト・フォーマットのもの）を出力する。

【0068】統合器は、入力されるMPEG-7 Access UnitのFragment Referenceに記述されている統合のための情報に基づき、MPEG-7 Access Unitを再統合し、MPEG-7記述を出力する。このとき、Fragment Reference内の下位構造化メタデータに関する参照情報における上位構造化メタデータへの統合の必要性を示す情報（属性use, Necessity）、及び／または、下位構造化メタデータの内容に関する情報に基づき、再統合の可否を判断するようにしてもよい。

【0069】一方、ビデオ複合化器は、入力されるビデオ符号化データを複合化し、ビデオ・コンテンツを出力する。

《インターネットを介したコンテンツ検索における属性の活用例》図18は、インターネットを介したビデオ・コンテンツ検索のネットワーク図である。ビデオデータベースには、多数のビデオ・コンテンツとそれぞれに対応するMPEG-7記述が記憶されている。また、MPEG-7記述は上位側MPEG-7記述であるビデオ・セグメント情報を含むAccess Unitと、下位側MPEG-7記述である色や音声などの分野ごとの記述子を含むAccess Unitとに上述の分割方法を用いて分割されている。

【0070】ビデオ・セグメント情報Access Unit内には複数のFragment Reference（接続位置情報と参照情報）が存在し、それぞれのFragment Referenceが分野ごとの各記述子Access Unitを参照すると共に、参照

先MPEG-7記述の内容の概略が属性として付加されている。ユーザーはインターネットを介して任意のMPEG-7 Access Unitをダウンロードし、ビデオ・コンテンツの検索を行うことができる。

【0071】図19は、図18に示したインターネットを介したビデオ・コンテンツ検索のフローチャートである。

【0072】ユーザーは、初めに、ビデオ・コンテンツ・データベースからビデオ・セグメント情報Access Unitをダウンロードする。次に、ビデオ・セグメント情報Access Unit内のFragment Referenceには参照先MPEG-7記述の内容の概略を示す属性が付加されているので、その情報を元にして検索に使用する記述子を含むAccess Unitをダウンロードする。例えば、色に関する記述子で検索するなら色に関する記述子Access Unitをダウンロードし、音声に関する記述子で検索するなら音声に関する記述子Access Unitをダウンロードする。また、検索に使用する記述子を含むAccess Unitを参照するFragment Referenceがそのビデオ・セグメント情報Access Unitに存在しなければ、そのコンテンツはスキップして次のコンテンツの検索に移る。

【0073】ダウンロードされたAccess Unit中のMPEG-7記述は、ユーザーが設定した検索条件と照らし合わされる。そして、もし検索条件に適合していれば、対応するビデオ・コンテンツをビデオ・コンテンツ・データベースからダウンロードし、検索を終了する。もし検索条件に適合しなければ、次のコンテンツの検索に移る。

【0074】次に、本発明の第4実施例について説明する。第4実施例では、MPEG-7記述の分割から蓄積・伝送までの処理の流れと、蓄積・伝送から再構成までの処理の流れは第1実施例と概略同様である。即ち、図1及び図2に示す処理の流れは、第4実施例にも当てはまる。但し、第4実施例では、Access Unitの構成が第1実施例のそれとは異なる。

【0075】図20は、本発明の第4実施例に用いる、Access Unitの構成例を示す概念図である。また、図21は、図20に示したAccess UnitのDDLシンタックスである。また、図22～図25は、図20に示したAccess Unitのバイナリ・フォーマットである。

【0076】第4実施例では、元のMPEG-7記述を1つの上位のMPEG-7記述と2つの下位のMPEG-7記述とに分割し、2つの下位のMPEG-7記述を1つのAccess Unitに格納した場合（即ち1つのAccess Unitに複数のMPEG-7記述を格納することができるようにした場合）を例に挙げて説明する。

【0077】第4実施例は、1つのAccess Unitに格納された複数の下位MPEG-7記述を、上位MPEG-7記述が格納されたAccess Unitから参照する場合に、効率的に参照できるようにしたものである。

【0078】Access Unitは、メタデータ格納部であるDescriptionと、接続先情報格納部であるFragment Referenceとを備えている。

【0079】Descriptionは、元のMPEG-7記述から分割された各MPEG-7記述を格納するものである。Descriptionは、1つのAccess Unit中に複数存在することも可能である。

【0080】Fragment Referenceは、Descriptionに格納されるMPEG-7記述とその下位のMPEG-7記述とを再接続する際に参照する接続先情報（接続位置情報と参照情報）を格納するものである。Fragment Referenceは、Connection Point（接続位置情報）とFragment Location（少なくとも下位MPEG-7記述の存在場所に関する情報を有する参照情報）との2つの子供要素を備えている。Connection PointとFragment Locationとの組は各下位MPEG-7記述毎に設けられる。第4実施例では下位MPEG-7記述が2つであるので、Connection PointとFragment Locationとの組は2組設けられる。

【0081】なお、Descriptionに格納される上位MPEG-7記述に接続されるべき下位のMPEG-7記述が1つも存在しない場合、Fragment Referenceも1つも存在しない。また、下位MPEG-7記述の格納状態に応じて、Fragment Referenceは1つのAccess Unit内に複数存在する場合もある。

【0082】Connection Pointは、Descriptionに格納される上位MPEG-7記述に対して下位MPEG-7記述を接続する位置に関する情報である。Connection Point中のXPathには、分割前の元のMPEG-7記述における下位MPEG-7記述の最上位ノードに相当するノード（下位MPEG-7記述のルート・ノードのパス（これはDescriptionに格納される上位MPEG-7記述には存在しないノード））を記述する。これにより、第1実施例で説明した属性positionがなくても、上位MPEG-7記述に対して下位MPEG-7記述を接続する位置を正確に記述することができる。

【0083】また、Connection Point中にelement Name（属性element Name）を設けてもよい。属性element Na

meには、分割前の元のMPEG-7記述における下位MPEG-7記述の最上位ノードに相当するノードの名前（下位MPEG-7記述のルート・ノードの名前）を記述する。これにより、分離の前後で下位MPEG-7記述のルート・ノードの名前が異なる場合に、再接続時に下位MPEG-7記述のルート・ノードの名前を分離前の名前に変換して接続することができる。

【0084】なお、ここでは、上位MPEG-7記述と下位MPEG-7記述とを1つの元のMPEG-7記述から分割したものという考え方をしたが、この考え方ではなく、MPEG-7記述同士を統合することを想定して、上位MPEG-7記述と下位MPEG-7記述とを個々に作成するという考え方もできる。この考え方に基づけば、Connection Point中のXPathには、上位MPEG-7記述と下位MPEG-7記述とが統合された場合に生成されると想定したMPEG-7記述における、前記下位MPEG-7記述の最上位ノードに相当するノード（下位MPEG-7記述のルート・ノードのパス）を記述することになる。属性element Nameには、前記想定したMPEG-7記述における下位MPEG-7記述の最上位ノードに相当するノードの名前（下位MPEG-7記述のルート・ノードの名前）を記述することになる。どちらの考え方をしても、Connection Point中のXPathに記述される内容は実質的には同一となる。属性element Nameに記述される内容も、どちらの考え方をしても、実質的には同一となる。

【0085】次に、Fragment Referenceのもう一方の子供要素であるFragment Locationについて説明する。Fragment Locationは、Descriptionに格納される上位MPEG-7記述に接続する下位MPEG-7記述を示す参照情報である。Fragment Locationのデータ型はuriReferenceである。

【0086】上位MPEG-7記述側のAccess Unitから下位MPEG-7記述を参照させる場合、基本的には、Fragment Location中のHRefに下位MPEG-7記述を含むAccess UnitのURI (UniformResource Identifiers)を記述する。

【0087】図20に示すように、上位MPEG-7記述に複数の下位MPEG-7記述を参照させ、さらにこれらの下位MPEG-7記述が同一のAccess Unitに含まれる場合、Fragment Reference内で各Connection Point及び各Fragment Locationとは独立した共通属性hrefに、そのAccess UnitのURIを記述する。そして、Fragment Refer

ence内に下位MPEG-7記述毎に存在する各Fragment Location中のHRefを省略してもよい。（各Fragment Location中のHRefに記述されるURIは共通の情報となるため。）

但し、同一のAccess Unitに含まれる複数の下位MPEG-7記述を一意に識別するために、各Fragment Location中のFragment Indexに複数の下位MPEG-7記述を識別する番号を記述する。番号は、Access Unitに含まれる下位MPEG-7記述に対して、先頭から順に正の整数を割り当てる。

【0088】また、参照する下位MPEG-7記述の概略内容を各Fragment Location内のcontent（属性content）に記述するようにしてもよい。複数の下位MPEG-7記述についての概略内容が同一の場合、Fragment Reference内で各Connection Point及び各Fragment Locationとは独立した共通属性contentに、その概略内容を記述する。そして、Fragment Reference中にある、該当する下位MPEG-7記述の各Fragment Locationの属性contentを省略してもよい。（各Fragment Location中の属性contentに記述される概略内容は共通の情報となるため。）また、複数の下位MPEG-7記述の概略内容が異なっても、その上位概念が同一の場合、Fragment Referenceの共通属性contentにその上位概念を記述し、そのFragment Reference中の各Fragment Locationの属性contentにはそれぞれの内容の概略を記述する。

【0089】以上により、Fragment Referenceの属性情報として、そのFragment Reference内の複数のFragment Locationに共通する情報が記述されていれば、個々のFragment Locationの情報を見なくても、その属性情報を見るだけで下位MPEG-7記述を含むAccess Unitを取得するか否かを判断したり、実際に取得したりできるので、参照情報の処理が簡潔になる。さらには、Fragment Referenceの属性情報として複数のFragment Locationに共通する情報を記述することに加えて、各Fragment Location中の該当する情報を省略すれば、参照情報の情報量が少なくなり、効率的に伝送できる。

【0090】また、複数のConnection Point中の属性element Nameに記述された下位MPEG-7記述のルート・ノードの名前が同一の場合、Fragment Reference中の各C

onnection Point及び各Fragment Locationとは独立した共通属性element Nameに、下位MPEG-7記述のルート・ノードの名前を記述し、各Connection Point中の属性element Nameを省略するようにしてもよい。これにより、接続位置情報の処理の簡潔化が図れ、接続位置情報の情報量の削減も可能となる。

【0091】このように、Fragment Reference (接続先情報)の属性情報をも上位MPEG-7記述と複数の下位MPEG-7記述との接続(統合)に利用することにより、Fragment Reference内の各Connection Point (接続位置情報)及び/または各Fragment Location (参照情報)の処理の簡潔化が図れ、接続(統合)の高速化が図れる。

【0092】もちろん、第4実施例は、第1実施例と同様に、複数のAccess Unitを取得したとき、接続先情報格納部であるFragment Referenceに格納された情報(接続位置情報と参照情報)を元にして、分離された上位MPEG-7記述と下位MPEG-7記述とを順次接続することにより、元のMPEG-7記述を再構成することができる。また、必要なAccess Unit、即ち必要なMPEG-7記述を参照情報に基づき選択することにより、元のMPEG-7記述を部分的に再構成することもできる。さらには、その伝送において、分割されたMPEG-7記述を各Access Unitとして伝送することにより、各Access Unit毎の伝送制御が行えるようになり伝送効率を高めることが可能となる。

【0093】また、接続のための情報である接続位置情報と参照情報とを、メタデータ格納部(Description)とは異なる接続先情報格納部(FragmentReference)に格納するようにしたので、構造化メタデータの内部を操作することなく接続のための情報を付加することができるという効果がある。

【0094】第4実施例においても、第1実施例と同様に、下位MPEG-7記述の統合の必要性を示す情報であるuse (属性use)をFragment Locationに記述してもよい。また、複数のAccess Unitを一意に識別するためのid (属性id: 識別情報)を格納した属性id格納部をAccess Unitに設けてもよい。

【0095】さらには、本実施例においても第3実施例で説明したように、Access Unit内のMPEG-7記述のデータ及び各情報(属性id、接続位置情報、参照情報、属性情報)をテキスト・データで表現してもよいし(図21参照)、Access Unit内のMPEG-7記述のデータ及び各情報をバイナリ・データで表現してもよい。図22は第4実施例におけるAccess Unitのバイナリ・フォーマット、図2

3はそのFragment Referenceのバイナリ・フォーマット、図24はそのConnection Pointのバイナリ・フォーマット、図25はそのFragment Locationのバイナリ・フォーマットである。

【0096】また、第4実施例は、図16に示すMPEG-7伝送システム、図17に示すMPEG-7受信システム、図18に示すビデオ・コンテンツ検索のネットワーク図、図19に示すビデオ・コンテンツ検索のフローチャートに当然適用可能である。

【0097】第4実施例では、元のMPEG-7記述を1つの上位のMPEG-7記述と2つの下位のMPEG-7記述とに分割した例を示したが、1つの上位のMPEG-7記述と5つの下位のMPEG-7記述とに分割した例を第5実施例(第4実施例と同様の分割、伝送、統合方法を用いるもの)として図26に示す。第5実施例では2つの下位のMPEG-7記述11, 12を格納した下位側第1のAccess Unitと、それぞれ1つずつの下位のMPEG-7記述21, 31, 41を格納した下位側第2～下位側第4のAccess Unitとが設けられているものとする。

【0098】上位のMPEG-7記述を格納したAccess Unitには、各下位MPEG-7記述毎に対応したConnection Point (接続位置情報)とFragment Location (参照情報)との組が設けられる。第5実施例では下位MPEG-7記述が5つであるので、Connection PointとFragment Locationとの組は5組設けられる。

【0099】各Fragment Location中のHRefには、第4実施例で説明したようにそれぞれ対応する下位MPEG-7記述を含むAccess UnitのURI (Uniform Resource Identifiers)を記述する。

【0100】図26に示すように、下位のMPEG-7記述11と下位のMPEG-7記述12とは同一の下位側第1のAccess Unitに格納されているので、下位のMPEG-7記述11, 12に対応する各Fragment Location中のHRefは共通の情報となる。そこで、下位のMPEG-7記述11, 12に対応するConnection PointとFragment Locationとの組を同一の第1のFragment Referenceに格納し、その第1のFragment Referenceの共通属性hrefに、下位側第1のAccess UnitのURIを記述する。この場合、Fragment Reference内に下位MPEG-7記述毎に存在する各Fragment Location中のHRefを省略してもよい。

【0101】下位側第1のAccess Unitに含

まれる2つの下位MPEG-7記述11, 12を一意に識別するために、それぞれの下位MPEG-7記述に対応する各Fragment Location中のFragment Indexに2つの下位MPEG-7記述11, 12を識別する番号を記述する。番号は、Access Unitに含まれる下位MPEG-7記述に対して、先頭から順に正の整数を割り当てる。

【0102】次に、下位側第2のAccess Unitに格納されている下位のMPEG-7記述21に対応するFragment Location内のcontent (属性content) と、下位側第3のAccess Unitに格納されている下位のMPEG-7記述31に対応するFragment Location内のcontent (属性content) とが同一内容であるとする。また、下位のMPEG-7記述21に対応するConnection Point中の属性element Nameと、下位のMPEG-7記述31に対応するConnection Point中の属性element Nameとが同一内容であるとする。

【0103】この場合、下位のMPEG-7記述21, 31に対応するConnection PointとFragment Locationとの組を同一の第2のFragment Referenceに格納する。そして、その第2のFragment Referenceの共通属性contentに、上記2つの属性contentの同一内容を記述すると共に、第2のFragment Referenceの共通属性element Nameに、上記2つの属性element Nameの同一内容を記述する。このとき、第2のFragment Reference内に下位MPEG-7記述毎に存在する各Fragment Locationの属性content、及び下位MPEG-7記述毎に存在する各Connection Pointの属性element Nameを省略してもよい。

【0104】また、下位MPEG-7記述21, 31の概略内容が異なっても、その上位概念が同一の場合、第2のFragment Referenceの共通属性contentにその上位概念を記述し、そのFragment Reference中の各Fragment Locationの属性contentにはそれぞれの内容の概略を記述するようにしてもよい。

【0105】下位側第4のAccess Unitに格納されている下位のMPEG-7記述41に対応するConnection PointとFragment Locationとは、他の下位のMPEG-7記述に対応するConnection PointとFragment Locationと共通内容がないので、第3のFragment Referenceに格納される。

【0106】なお、上記説明では、構造化メタデータを、コンテンツの内容を記述するための木構造を有する構造化メタデータとしたが、電子番組案内を記述するための木構造を有する構造化メタデータとしてもよい。電子番組案内とは一般的にEPGと呼ばれているものであり、番組タイトル、放送日時、放送チャンネル、番組概要等の情報である。

【0107】また、上述の各実施例で説明した構造化メタデータ (MPEG-7記述) の分割方法、統合方法を実行させるためのプログラムを、インターネット、CD-ROM等で各ユーザに配布し、各ユーザはそのプログラムをパソコン等に取り込み、そのプログラムを取り込んだ機器を分割装置、統合装置として用いてもよい。

【0108】

【発明の効果】以上の通り、本発明の構造化メタデータの伝送方法は下記の効果を有する。

(イ) 統合可能なように複数の構造化メタデータをユニット化し、各構造化メタデータを各ユニットとして伝送することにより、各ユニット毎の伝送制御が行えるようになり、伝送効率を高めることができる。さらに本発明を用いれば、受信側において欠落データが発生した場合にも、不完全となる構造化メタデータの大きさを小さくと定めることができると共に、不完全な構造化メタデータのみを再度補間できる。

(ロ) 各ユニットにそれぞれのユニットを特定するための識別情報を付加して前記各ユニットを伝送するようにしたので、ディジタル・テレビジョン放送等の伝送システムにおいても、各ユニットを一意に識別することができる。

(ハ) 本発明を用いれば、上位階層の情報、下位階層の情報といった、コンテンツの内容記述の詳細度に応じた伝送が行える。

(ニ) 第1の構造化メタデータと第2の構造化メタデータとの接続のための情報である接続位置情報と参照情報とを、メタデータ格納部とは異なる接続先情報格納部に格納するようにしたので、第1の構造化メタデータの内部を操作することなく接続のための情報を付加して伝送できる。

(ホ) 第1のユニットにおいて、第1の構造化メタデータ内における第2の構造化メタデータの接続点を示す接続位置情報が、第1の構造化メタデータ内におけるノードを指定する情報と、そのノードに対する第2の構造化メタデータの接続位置を指定する情報とを有するものとした場合には、接続位置情報における接続点を正確に指定する記述が容易となる。

(ヘ) 第1のユニットにおいて、第2の構造化メタデータに関する参照情報が、第2の構造化メタデータの内容に関する情報を有する場合には、接続先の第2の構造化メタデータを入手することなく接続先メタデータの内容の概略を知ることができ、接続先の第2の構

造化メタデータを入手する必要があるかないかを、接続先の構造化メタデータを入手することなく判断できる。

(ト) 第1のユニットにおいて、第2の構造化メタデータに関する参照情報が、第1の構造化メタデータへの統合（接続）の必要性を示す情報を有する場合には、統合先（接続先）の構造化メタデータを入手することを強制できる。統合先（接続先）の構造化メタデータを著作権保護データとしておけば、統合元（接続元）の構造化メタデータの著作権保護を強化できる。

(チ) ユニットを特定するための識別情報、前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接続位置情報、及び前記第2の構造化メタデータに関する参照情報を、テキスト形式で記述した場合には、ユーザーにとって理解が容易である、編集が容易である、インターネットでのデータの送受信に好適である等の利点を有する。

(リ) ユニットを特定するための識別情報、前記第1の構造化メタデータ、前記第2の構造化メタデータ、前記接続位置情報、及び前記第2の構造化メタデータに関する参照情報を、バイナリ形式で記述した場合には、テキスト・データで表現された各情報よりも情報量が少なく伝送効率を向上させることができるので、デジタル・テレビジョン放送、携帯端末向けのデータ配信等でのデータの伝送に好適である。

(ヌ) 前記第1のユニットにおいて、接続位置情報を、分割前の前記構造化メタデータにおける（または、前記第1の構造化メタデータと前記第2の構造化メタデータとが統合された場合に生成されると想定した木構造を有する第3の構造化メタデータにおける）、前記第2の構造化メタデータの最上位ノードに相当するノードを示す情報とした場合には、前記第1の構造化メタデータに対して前記第2の構造化メタデータを接続（統合）する位置をより少ない情報量でより正確に記述することができる。

(ル) 前記第1のユニットにおいて、前記接続位置情報が、分割前の前記構造化メタデータにおける（または、前記第1の構造化メタデータと前記第2の構造化メタデータとが統合された場合に生成されると想定した木構造を有する第3の構造化メタデータにおける）、前記第2の構造化メタデータの最上位ノードに相当するノードの名前に関する名前情報を有する場合には、前記第2の構造化メタデータの最上位ノードの名前が分離前の前記構造化メタデータにおける（または前記想定した第3の構造化メタデータにおける）そのノードに相当するノードの名前と異なる場合にも、接続時（統合時）に、前記第2の構造化メタデータの最上位ノードの名前を分離前の名前（または前記想定した第3の構造化メタデータにおける名前）に変換して接続（統合）することができる。

(ロ) 2つ以上の第2の構造化メタデータが存在する

場合において、各前記第2の構造化メタデータの各接続位置情報における2つ以上の前記接続位置情報に共通する情報である共通接続位置情報と、各前記第2の構造化メタデータに関する各参照情報における2つ以上の前記参照情報に共通する情報である共通参照情報との、2つの共通情報の内の少なくとも一方の共通情報を、前記第1のユニット内の接続先情報格納部に属性共通情報として有する場合には、前記属性共通情報の有する情報に応じて、前記属性共通情報を、前記接続位置情報及び／または前記参照情報に対して優先して、前記第1及び第2の構造化メタデータの統合時に用いることにより、前記第1のユニット内の接続先情報格納部にある各接続位置情報及び／または各参照情報の処理の簡潔化が図れ、接続（統合）時の高速化が図れる。

【図面の簡単な説明】

【図1】第1実施例を用いた構造化メタデータの分割方法、伝送方法、及び統合方法の内の、分割から蓄積・伝送までの処理の流れの概略を示す図である。

【図2】第1実施例を用いた構造化メタデータの分割方法、伝送方法、及び統合方法の内の、蓄積・伝送から再構成までの処理の概略を示す図である。

【図3】第1実施例に用いるAccess Unitの構成を示す概念図である。

【図4】図3に示したAccess UnitのDDLシンタックス例である。

【図5】第2の実施例に用いるAccess Unitの構成を示す概念図である。

【図6】図5に示したAccess UnitのDDLシンタックス例である。

【図7】図5に示したAccess UnitのDDLシンタックス例である。

【図8】属性positionの使用方を説明するための図である。

【図9】属性positionの使用方を説明するための図である。

【図10】属性positionの使用方を説明するための図である。

【図11】属性positionの使用方を説明するための図である。

【図12】属性positionの使用方を説明するための図である。

【図13】属性positionの使用方を説明するための図である。

【図14】第3実施例におけるAccess Unitのバイナリ・フォーマットを示す図である。

【図15】第3実施例におけるFragment referenceのバイナリ・フォーマットを示す図である。

【図16】MPEG-7伝送システムを示す図である。

【図17】MPEG-7受信システムを示す図である。

【図18】一実施例を用いたインターネットを介してのビデオ・コンテンツ検索のネットワークを示す図である。

【図19】インターネットを介したビデオ・コンテンツ検索の処理の流れを示すフローチャートである。

【図20】第4実施例に用いるAccess Unitの構成を示す概念図である。

【図21】図20に示したAccess UnitのDDLシンタックス例である。

【図22】第4実施例におけるAccess Unit

のバイナリ・フォーマットである。

【図23】第4実施例におけるFragment Referenceのバイナリ・フォーマットである。

【図24】第4実施例におけるConnection Pointのバイナリ・フォーマットである。

【図25】第4実施例におけるFragment Locationのバイナリ・フォーマットである。

【図26】第5実施例に用いるAccess Unitの構成を示す概念図である。

【図4】

図4

```
<element name="AccessUnit" type="mpeg7:AccessUnitType"/>
<complexType name="AccessUnitType">
  <sequence>
    <element name="FragmentReference"
      type="mpeg7:FragmentReferenceType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="Description" type="mpeg7:DescriptionType"/>
  </sequence>
  <attribute name="id" type="ID"/>
</complexType>

<complexType name="FragmentReferenceType">
  <element name="ConnectionPoint" type="mpeg7:ConnectionPointType"/>
  <element name="FragmentLocation" type="mpeg7:FragmentLocationType"/>
</complexType>

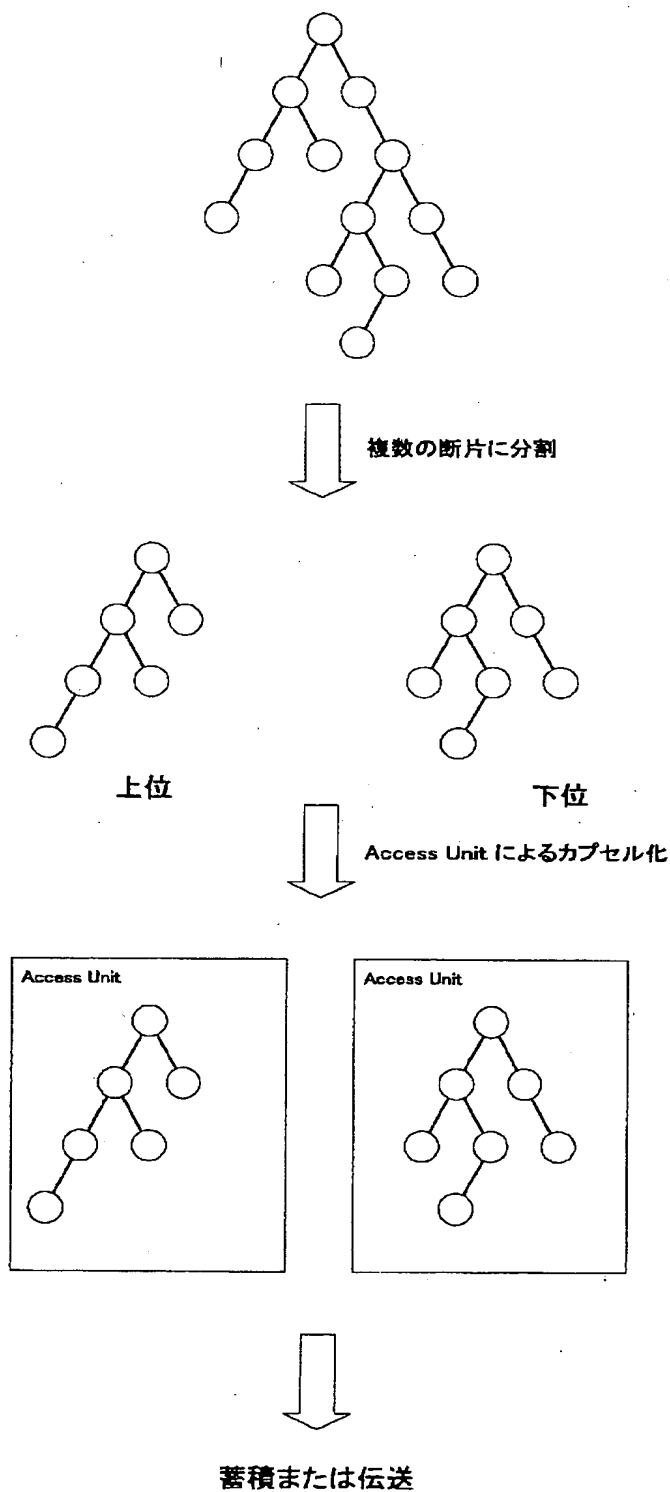
<complexType name="ConnectionPointType">
  <choice>
    <element name="IDRef" type="IDREF"/>
    <element name="XPath" type="mpeg7:xPathType"/>
  </choice>
  <attribute name="position" use="default" value="after">
    <restriction base="string">
      <enumeration value="before"/>
      <enumeration value="after"/>
      <enumeration value="first"/>
      <enumeration value="last"/>
    </restriction>
  </attribute>
</complexType>

<complexType name="FragmentLocationType">
  <simpleContent>
    <extension base="uriReference">
      <attribute name="content" type="Name"/>
      <attribute name="use" use="default" value="optional">
        <restriction base="string">
          <enumeration value="optional"/>
          <enumeration value="required"/>
        </restriction>
      </attribute>
    </extension>
  </simpleContent>
</complexType>

<complexType name="DescriptionType">
  <any namespace="##any" processContent="skip" minOccurs="1"
    maxOccurs="unbounded"/>
</complexType>
```

【図1】

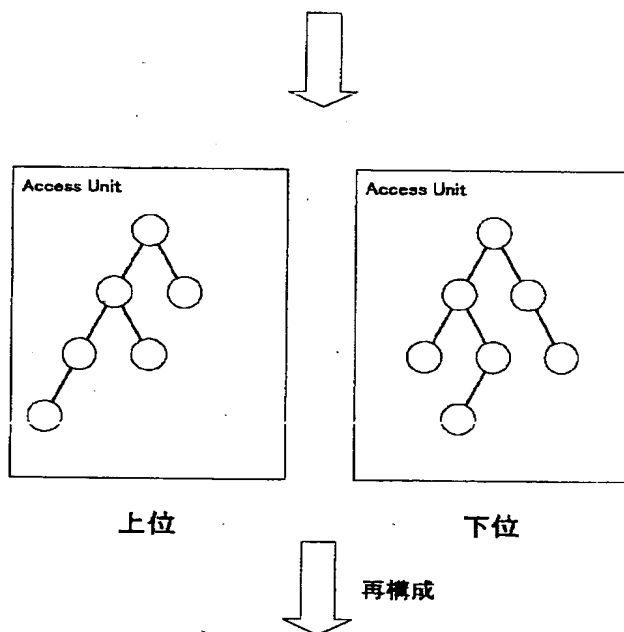
図1



【図2】

図2

蓄積または伝送



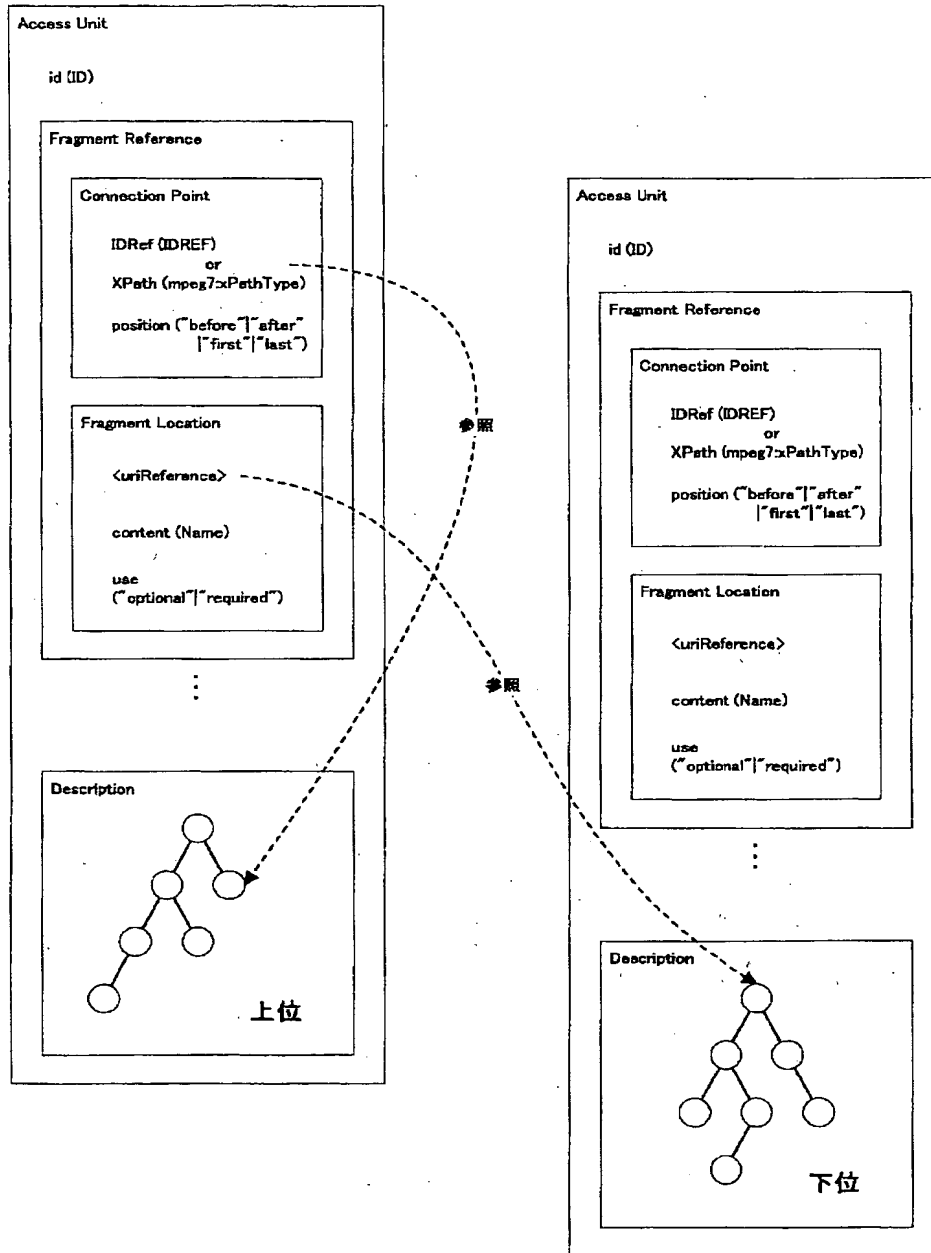
【図14】

図14

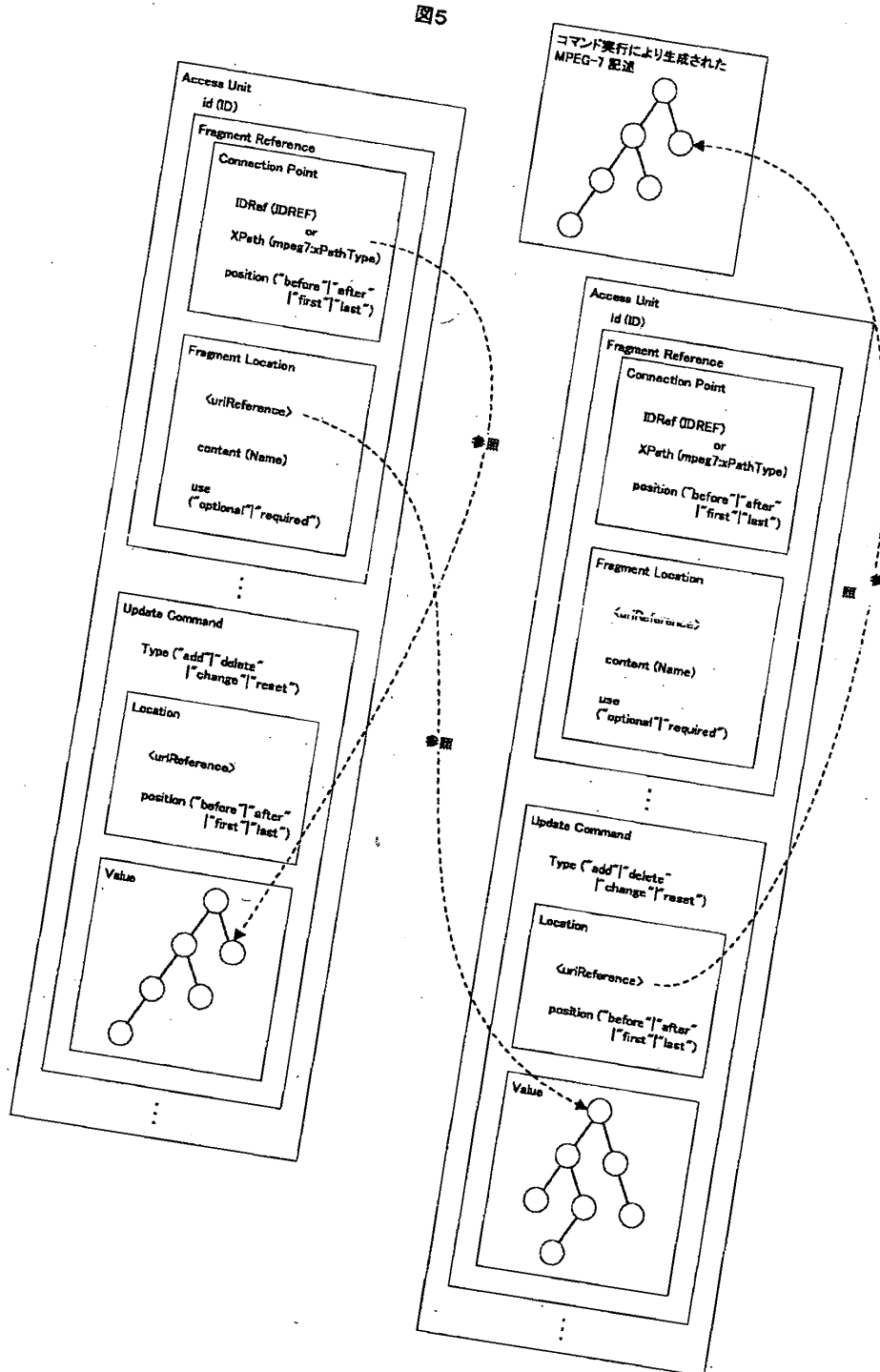
| | | | | | | |
|------|-------------------------------|-------------------------------|--------------------|--------------------|-----|--------------------------------|
| AUID | Length of fragment references | Number of fragment references | Fragment reference | Fragment reference | ... | Sub-tree binary representation |
|------|-------------------------------|-------------------------------|--------------------|--------------------|-----|--------------------------------|

【図3】

図3



【図5】



【図6】

図6

```

<element name="AccessUnit" type="mpeg7:AccessUnitType"/>
<complexType name="AccessUnitType">
  <sequence>
    <element name="FragmentReference"
      type="mpeg7:FragmentReferenceType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="UpdateCommand" type="mpeg7:UpdateCommandType"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="id" type="ID"/>
</complexType>
<complexType name="FragmentReferenceType">
  <element name="ConnectionPoint" type="mpeg7:ConnectionPointType"/>
  <element name="FragmentLocation" type="mpeg7:FragmentLocationType"/>
</complexType>
<complexType name="ConnectionPointType">
  <choice>
    <element name="IDRef" type="IDREF"/>
    <element name="XPath" type="mpeg7:xPathType"/>
  </choice>
  <attribute name="position" use="default" value="after">
    <restriction base="string">
      <enumeration value="before"/>
      <enumeration value="after"/>
      <enumeration value="first"/>
      <enumeration value="last"/>
    </restriction>
  </attribute>
</complexType>
<complexType name="FragmentLocationType">
  <simpleContent>
    <extension base="uriReference">
      <attribute name="content" type="Name"/>
      <attribute name="use" use="default" value="optional">
        <restriction base="string">
          <enumeration value="optional"/>
          <enumeration value="required"/>
        </restriction>
      </attribute>
    </extension>
  </simpleContent>
</complexType>
<complexType name="UpdateCommandType">
  <sequence>
    <element name="Type" type="mpeg7:UpdateTypeType"/>
    <element name="Location" type="mpeg7:UpdateLocationType"/>
    <element name="Value" type="mpeg7:UpdateValueType"
      minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>

```

【図15】

図15

| Reference of content ID | Necessity | Reference to AUID | Connection point |
|----------------------------|-----------|----------------------|---------------------|
|----------------------------|-----------|----------------------|---------------------|

【図7】

図7

```

<simpleType name="UpdateTypeType">
  <restriction base="string">
    <enumeration value="add"/>
    <enumeration value="delete"/>
    <enumeration value="change"/>
    <enumeration value="reset"/>
  </restriction>
</simpleType>

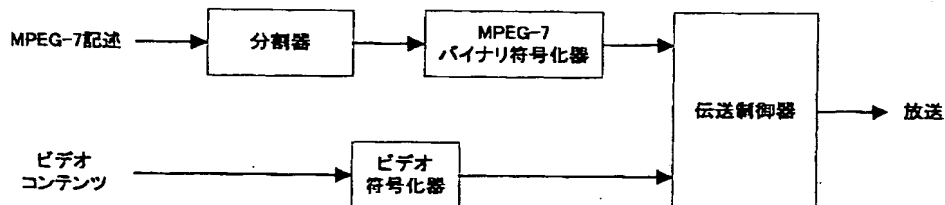
<complexType name="UpdateLocationType">
  <simpleContent>
    <extension base="uriReference">
      <attribute name="position" use="default" value="after">
        <restriction base="string">
          <enumeration value="before"/>
          <enumeration value="after"/>
          <enumeration value="first"/>
          <enumeration value="last"/>
        </restriction>
      </attribute>
    </extension>
  </simpleContent>
</complexType>

<complexType name="UpdateValueType">
  <any namespace="##any" processContent="skip" minOccurs="1"
    maxOccurs="unbounded"/>
</complexType>

```

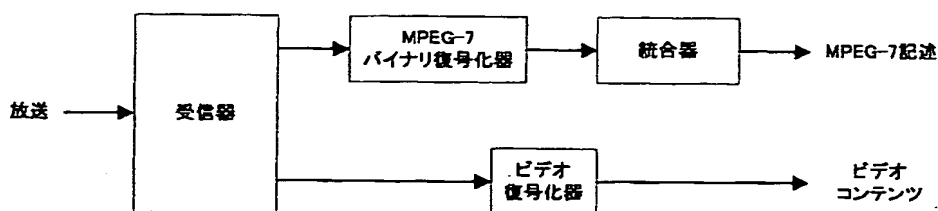
【図16】

図16



【図17】

図17

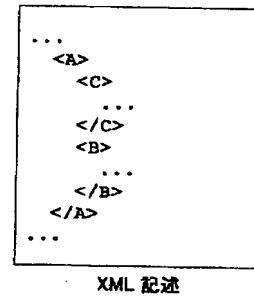
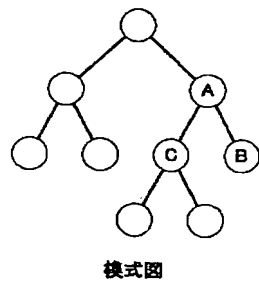
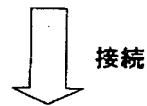
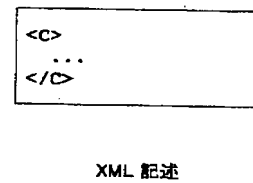
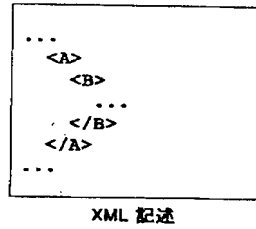
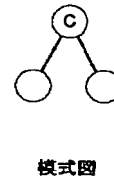
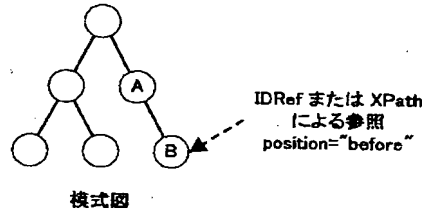


【図8】

図8

上位 MPEG-7 記述

下位 MPEG-7 記述

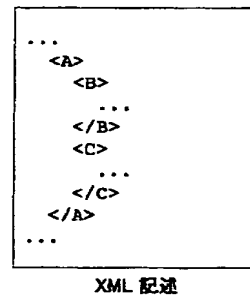
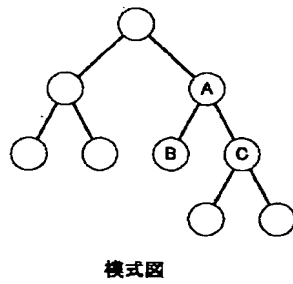
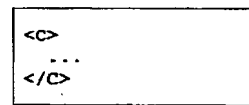
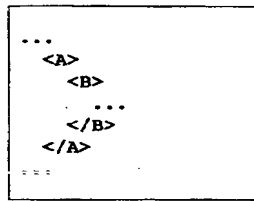
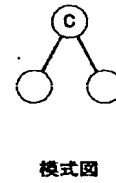
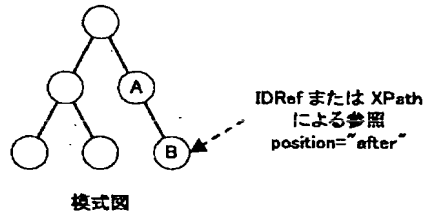


【図 9】

図9

上位 MPEG-7 記述

下位 MPEG-7 記述

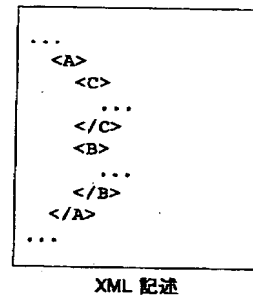
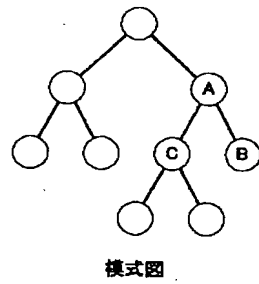
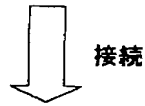
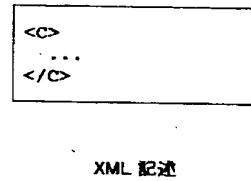
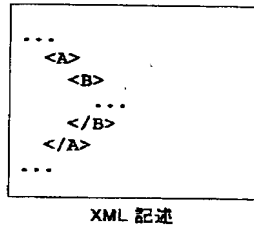
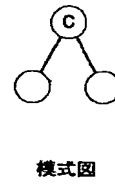
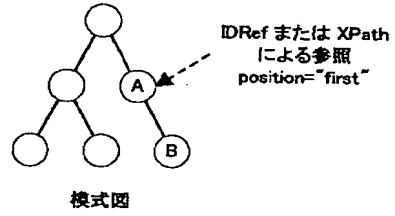


【図10】

図10

上位 MPEG-7 記述

下位 MPEG-7 記述

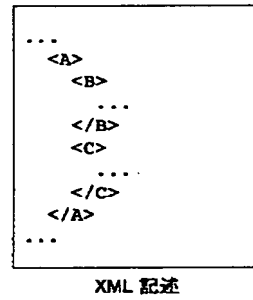
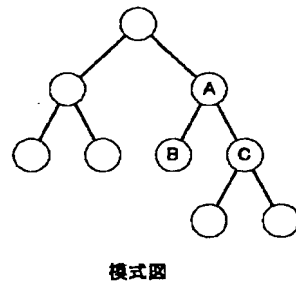
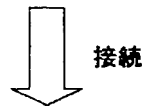
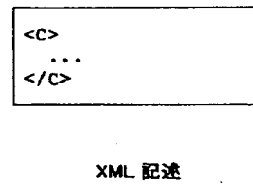
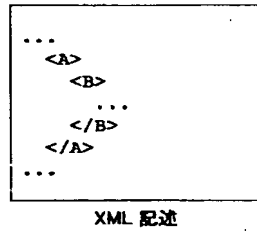
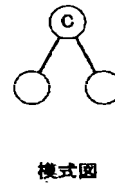
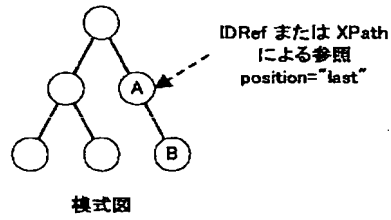


【図11】

図11

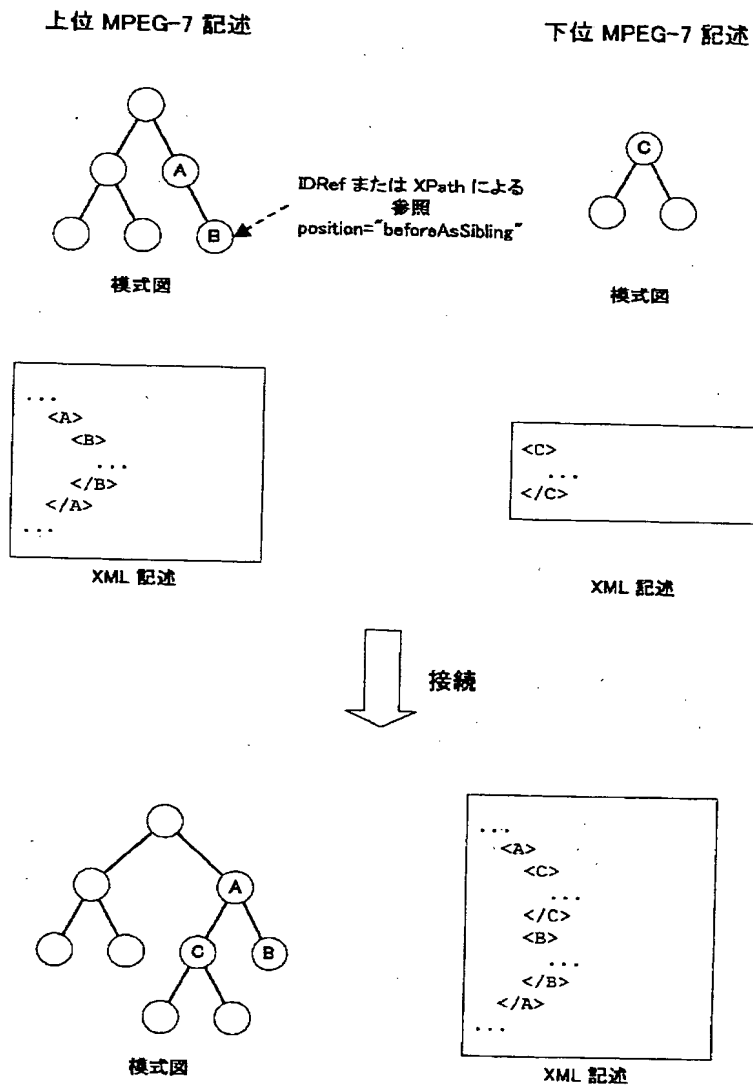
上位 MPEG-7 記述

下位 MPEG-7 記述



【図12】

図12

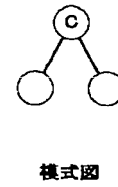
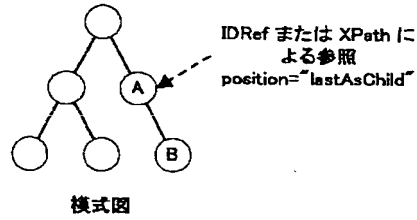


【図13】

図13

上位 MPEG-7 記述

下位 MPEG-7 記述



```

...
<A>
  <B>
    ...
  </B>
</A>
...

```

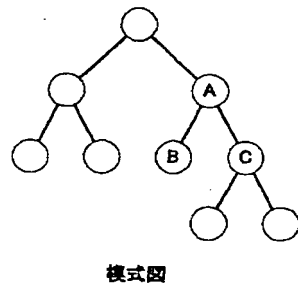
XML 記述

```

<C>
  ...
</C>

```

XML 記述



```

...
<A>
  <B>
    ...
  </B>
  <C>
    ...
  </C>
</A>
...

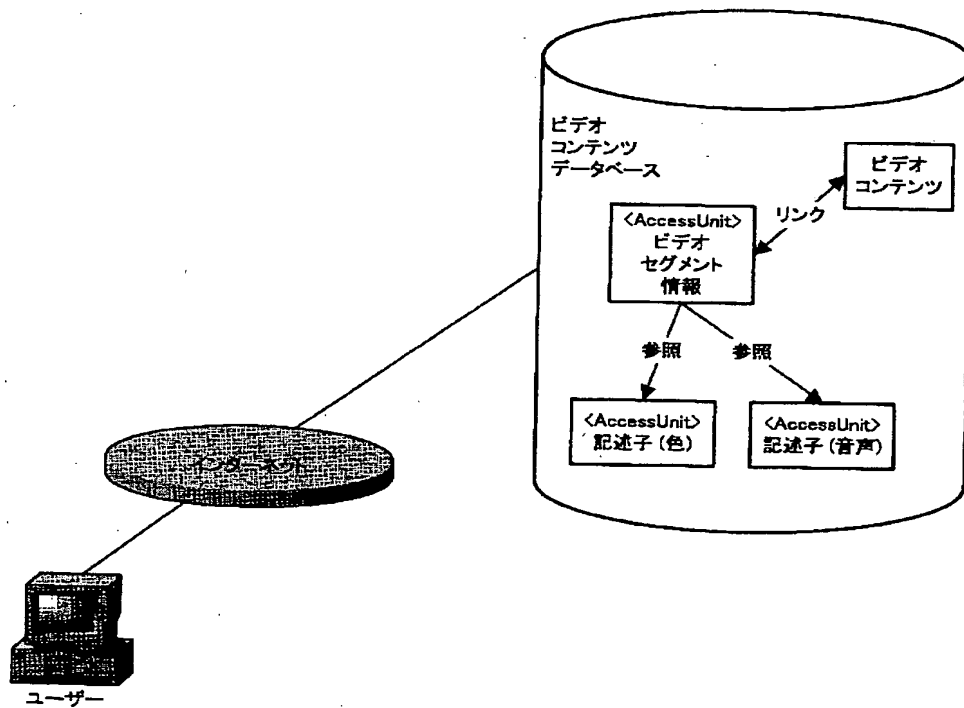
```

XML 記述

【図18】

図18

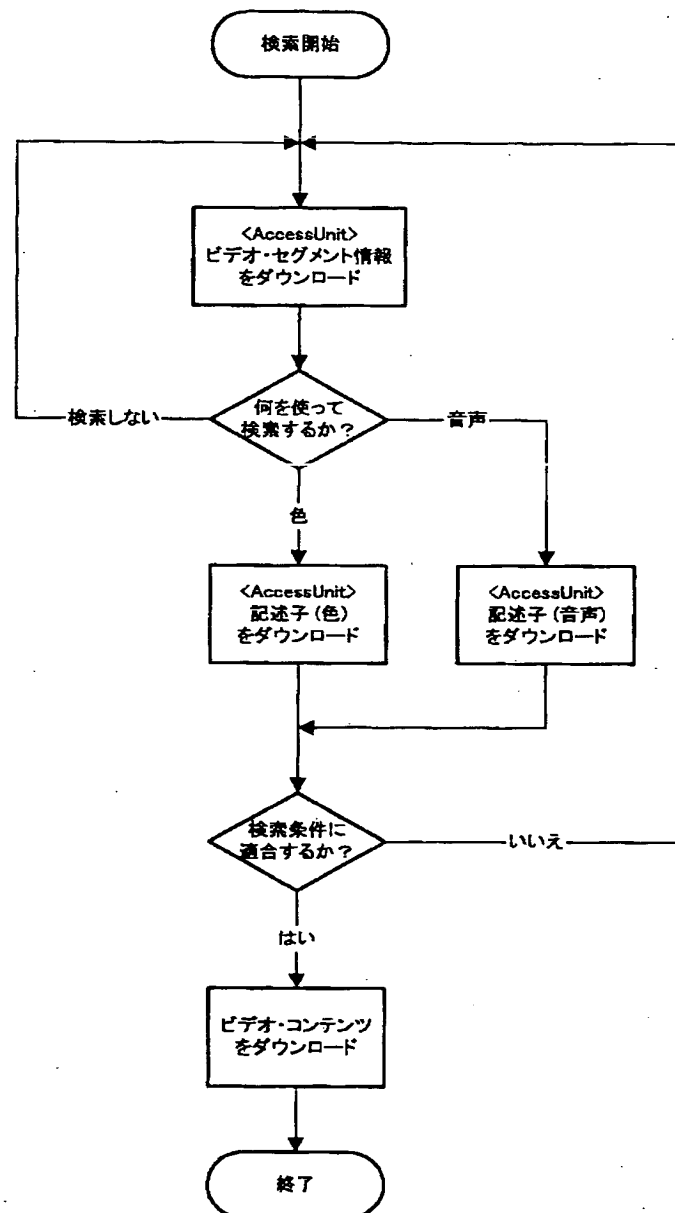
インターネットを介したビデオ・コンテンツ検索のネットワーク図



【図19】

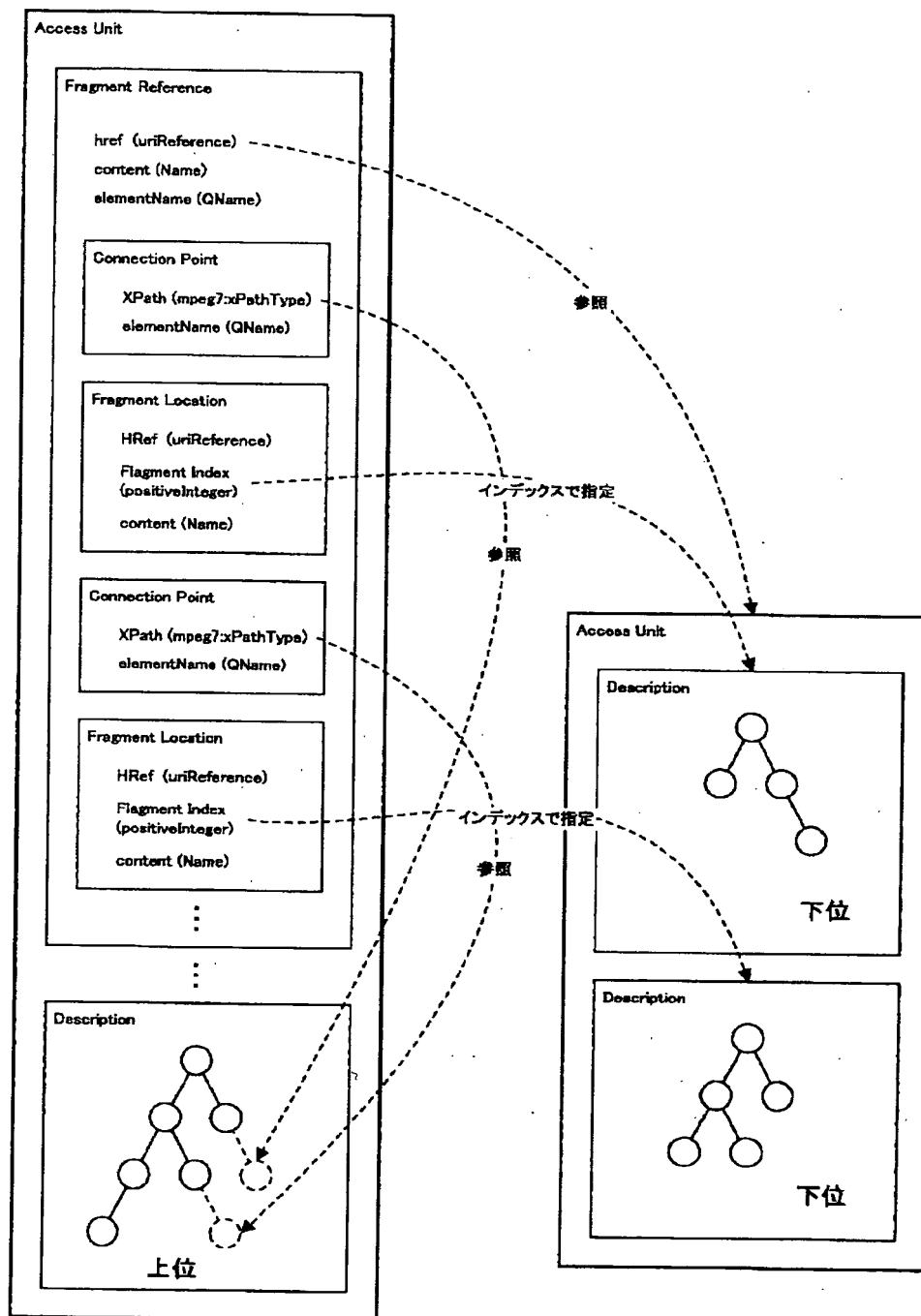
図19

インターネットを介したビデオ・コンテンツ検索のフローチャート



【図20】

図20



【図21】

図21

```

<element name="AccessUnit" type="mpeg7:AccessUnitType"/>
<complexType name="AccessUnitType">
  <sequence>
    <element name="FragmentReference"
      type="mpeg7:FragmentReferenceType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="Description" type="mpeg7:DescriptionType"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="FragmentReferenceType">
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="ConnectionPoint"
      type="mpeg7:ConnectionPointType"/>
    <element name="FragmentLocation"
      type="mpeg7:FragmentLocationType"/>
  </sequence>
  <attribute name="href" type="uriReference" use="optional"/>
  <attribute name="content" type="Name" use="optional"/>
  <attribute name="elementName" type="QName" use="optional"/>
</complexType>
<complexType name="ConnectionPointType">
  <sequence>
    <element name="XPath" type="mpeg7:xPathType"/>
  </sequence>
  <attribute name="elementName" type="QName" use="optional"/>
</complexType>
<complexType name="FragmentLocationType">
  <sequence>
    <element name="ERef" type="uriReference" minOccurs="0"/>
    <element name="FragmentIndex" type="positiveInteger"
      minOccurs="0"/>
  </sequence>
  <attribute name="content" type="Name" use="optional"/>
</complexType>
<complexType name="DescriptionType">
  <any namespace="##any" processContent="skip" minOccurs="1"
    maxOccurs="unbounded"/>
</complexType>

```

【図22】

図22

| AccessUnit { | Number of bits | Mnemonic |
|---|-------------------|----------|
| FragmentReferenceFlag | 1 | bslbf |
| if (FragmentReferenceFlag == 1) { | | |
| do { | | |
| FragmentReference() | | |
| NextFragmentReferenceFlag | 1 | bslbf |
| } while (NextFragmentReferenceFlag == 1) | | |
| } | | |
| do { | | |
| Description() | | |
| NextDescriptionFlag | 1 | bslbf |
| } while (NextDescriptionFlag == 1) | | |
| } | | |

【図23】

図23

| FragmentReference { | Number of bits | Mnemonic |
|--|---------------------|----------|
| contentFlag | 1 | bslbf |
| if (contentFlag == 1) { | | |
| content | | bslbf |
| } | | |
| hrefFlag | 1 | bslbf |
| if (hrefFlag == 1) { | | |
| hrefLength | 8 | uimsbf |
| href | 8*hrefLength | bslbf |
| } | | |
| elementNameFlag | 1 | bslbf |
| if (elementNameFlag == 1) { | | |
| elementNameLength | 8 | uimsbf |
| elementName | 8*elementNameLength | bslbf |
| } | | |
| do { | | |
| ConnectionPoint() | | |
| FragmentLocation() | | |
| NextConnectionPointFlag | 1 | bslbf |
| } while (NextConnectionPointFlag == 1) | | |
| } | | |

【図24】

図24

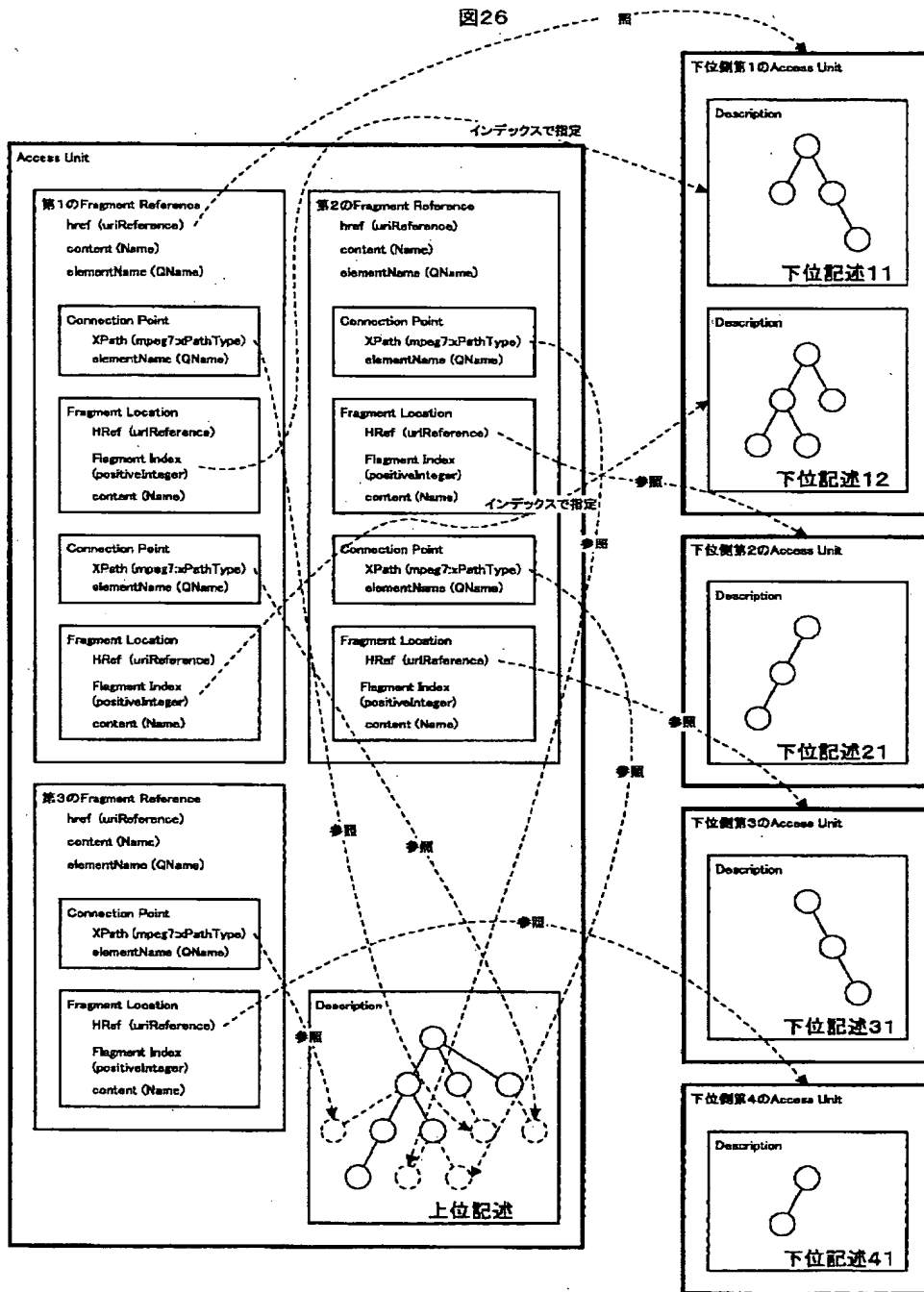
| ConnectionPoint { | Number of bits | Mnemonic |
|-------------------------------|---------------------|----------|
| elementNameFlag | 1 | bslbf |
| if (elementNameFlag == 1) { | | |
| elementNameLength | 8 | uimsbf |
| elementName | 8*elementNameLength | bslbf |
| } | | |
| XPathLength | 8 | uimsbf |
| XPath | 8*XPathLength | bslbf |
| } | | |

【図 25】

図 25

| FragmentLocation { | Number of bits | Mnemonic |
|------------------------|-------------------|----------|
| HRefFlag | 1 | bslbf |
| if (HRefFlag == 1) { | | |
| HRefLength | 8 | uimsbf |
| HRef | 8*HRefL ength | bslbf |
| } | | |
| FragmentIndex | 5 - infinite | vuimsbf |
| } | | |

〔図26〕



フロントページの続き

| | | | |
|----------------------------|-------|---------|--------------|
| (51) Int. Cl. ⁷ | 識別記号 | F I | ターマコード* (参考) |
| H 0 4 N | 7/08 | H 0 4 N | 7/13 |
| | 7/081 | | 7/08 |
| // H 0 4 H | 1/00 | | |

Fターム(参考) 5B075 KK07 KK25 KK35 ND16 NS10
5B082 BA09 HA05 HA08
5C059 MA00 PP20 RC05 RC12 RC19
RC24 RC34 RF23 SS02 SS11
UA06 UA39
5C063 AB03 AB05 CA29 CA34 CA36
DA03 DA07 EB01 EB27 EB39

